SRI International

THE VISION PROBLEM: EXPLOITING PARALLEL COMPUTATION

Technical Note No. 458

February 28, 1989

By: **Martin A. Fischler, Oscar Firschein, Stephen T. Barnard, Pascal V. Fua, and Yvan Leclerc**

Artificial Intelligence Center
Computer and Information Sciences Division

SRI Projects 2000 and 8388

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract Nos. MDA903-86-C-0084 and DACA76-85-C-0004.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
28 FEB 1989		**00-02-1989 to 00-02-1989**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Vision Problem: Exploiting Parallel Computation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**68**	

Abstract

This technical report consists of an introductory paper and three technical papers presented at the session, "AI Application of Supercomputers: The Vision Problem," at the Fourth International Conference on Supercomputing, Santa Clara, California, April 30 to May 5, 1989.

AI Application of Supercomputers: The Vision Problem

Oscar Firschein and Martin A. Fischler

Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

February 21, 1989

Abstract

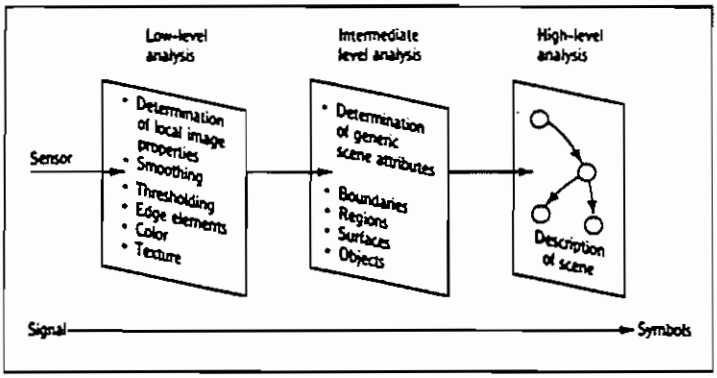
For the session "AI Application of Supercomputers: The Vision Problem," in the Fourth International Conference on Supercomputing, the major problems in computer vision are outlined, and the "signals to symbols" (SS) and the "monolithic computing"(MC) approaches to these problems are described. We note that the availability of parallel computation makes the MC approach feasible.*

Introduction

Computer Vision, as a scientific endeavor, can be viewed as an attempt to solve a set of specific problems concerned with deducing the nature of the surrounding environment from imaged data. The most prominent of these problems are:

- Modeling scene geometry
- Detecting and delineating significant scene structures (perceptual grouping)
- Naming or labeling the detected scene structures

*Support for this work was provided by the Defense Advanced Research Projects Agency under contracts DCA76-85-C-0004 and MDA903-86-C-0084.



Raw sensed data are transformed into a description of the scene by a series of inductive steps.

Figure 1: The signals-to-symbols paradigm for computational vision

A number of additional issues and problems must be addressed to build practical and effective vision systems. These include the need for:

- An architectural concept for integrating the components of the system
- A way of internally representing prior knowledge and the instantiated models (what knowledge do we store, and in what form do we store it)
- A way of communicating with the outside world and, especially, a way of allowing a human operator to visualize the environmental model constructed by the machine

Two major architectural paradigms have evolved for dealing with the first three problems. The dominant paradigm, *signals-to-symbols* (SS), views the visual process as the construction of a layered set of representations that describe the external world with successively more semantically oriented and globally referenced concepts. As shown in Figure 1, at the lowest levels of the "interpretation pyramid" (i.e., at the levels closest to the sensor), "local" processes act on an *image* or image pair, to delineate coherent regions and intensity events (e.g., edges), to find correlations (e.g., disparities) between points or regions in one or more of the images, and to provide measurements over an image of such point properties as color and texture. Processes at an intermediate level employ context and generic (world) knowledge to recover geometric (e.g., surface depth and orientation, object boundaries) and photometric (e.g., shadows, surface color) information about the *scene*. At the highest level, that of semantic interpretation, either explicit names or class labels are assigned to the delineated scene objects.

The second major architectural concept, which can be termed the *monolithic computation* (MC) paradigm, is based on the idea that the relation between image data and the scene model is best described by an objective function (to be optimized), or by a set of constraints or equations (to be satisfied); the intermediate states in the computational process do not necessarily have any obvious semantic meaning (as they do in the signals-to-symbols paradigm). Variants of the MC paradigm include signal theory, statistical decision theory, connectionism, and neural-net approaches.

Neither SS nor MC has as yet allowed its adherents to construct a general-purpose vision system that can successfully operate in the outdoor world. Some of the main reasons for this state of affairs are given below (more detailed discussions are presented elsewhere [3, 4]):

- Neither approach has evolved effective mechanisms for linking measurable image attributes to the function, purpose, or intent of the scene objects to be identified. Thus, recognition must be based on immediate appearance — which is too restricted a base for identifying many of the objects we are typically interested in.

- Neither approach has provided an effective way to describe or represent complex objects or object classes (e.g., vegetation), the difference in appearance between members of such classes (e.g., between cats and dogs), or flexible or articulated objects (people, water, a piece of string), or "formless" objects (a crumpled piece of paper, a sweater lying in a heap, a rock). Thus, recognition is typically limited to objects that can be described by simple or explicit shape models, or that have relatively unique attributes (e.g., a distinguishing color).

- Neither approach offers a prescription for actively interacting with the surrounding environment in acquiring sensory information, or even for effectively using the sequence of images returned by a sensor moving through its environment. Thus, low-level analysis is usually based on the information contained in a single image or, at best, the correlations between a pair of images.

One major difference between SS and MC is that SS is inherently "local" in its processing of low-level information — it has no effective way of taking advantage of context, and thus of being able to judge the "correctness" of the low-level decisions it makes, or to properly set the inevitable parameters controlling the behavior of the low-level analysis techniques. Theoretically, feedback from the higher levels (in SS) should be able to deal with the above problems, but in practice, no such solution has yet been demonstrated to be effective. A major advantage of MC is that it provides a natural way of employing "context" because it deals with all of its information at once, rather than partitioning the analysis into a series of relatively independent subproblems. Partitioning, of course, is a way of dealing with computational complexity, and MC techniques generally require immense computational resources for real-time operation.

The Role of Parallel Computation in MC

Our current work at SRI International addresses the above problems. A major theme of our research is an attempt to understand how to effectively use various forms of optimization and global analysis (at least within semantically coherent portions of the modeling task) — i.e., a move toward the MC end of the spectrum. The three papers presented at this session use an MC approach in the context of the stereo vision problem, the scene partitioning problem, and the problem of recognizing cultural features.

Stereo vision. Barnard [1] has developed a stereo algorithm (Hierarchical Stochastic Stereo, or HSS) that embeds local matching of individual pixels in a global optimization framework. The approach uses stochastic techniques to optimize an objective function that rewards correspondences between pixels that are similar in intensity value and whose disparities are similar to those of their neighboring pixels. "Simulated annealing" over a hierarchy of images is used to find the complete set of correspondences that best satisfies the objective function. Because individual pixels (rather than finite areas) are matched, projective distortion is no longer a problem and the system can exploit the full resolution of the digitized image rather than the effectively reduced resolution created by a correlation "patch." Experiments show that this approach can successfully compile a dense depth model of natural three-dimensional scenes. We have used this technique to produce dense elevation maps of a test site outside of Denver — we can map on 0.3-meter centers as compared with 5 meters in the best previously available digital terrain maps.

Scene partitioning. A paper by Leclerc [5] introduces and formalizes an optimization-based approach, applicable to both image partitioning and subsequent steps in the scene analysis process, which involves finding the "best" description of an image in terms of some specified descriptive language. Leclerc employs a language that describes the image in terms of regions having a low-order polynomial intensity variation plus white noise; region boundaries are described by a differential chain code. A continuation method is used to find a best description, in the sense of least encoding length, that is both stable (i.e., minor perturbations in the viewing conditions should not alter the description) and complete (i.e., the image, including any noise or errors, must be completely explained by the description).

Recognizing cultural features. Fua and Hanson [2] have proposed to extract generic shapes from monoscopic and stereoscopic imagery based on generating shape hypotheses and ranking them according to a model-based measure. As in the case of Leclerc, they show that this process finds the "best" description of the scene in terms of the shape models used, and can be understood as the optimization of an objective function. Because the search space is extremely large, their implementation on a conventional computer uses the SS paradigm to guide the search. Using a parallel machine, they can now investigate methods for performing the optimization in the

MC framework. In their recent work, they explore ways to outline object contours automatically, given a very rough estimate of the location of the object. This is done by modeling object boundaries as smooth curves and deforming the curves so as to optimize the fit to the image data. The fit is based on a model requiring smooth intensity variation within bounding contours, and preservation of contour shape when multiple images are available. This technique is powerful because it uses the image information to its full extent, but a large amount of computation is required at every step of the optimization process.

All of these approaches are made practical only because of the availability of parallel computation. We expect fast parallel computers to exert an important influence on the solution of some of the basic vision problems.

References

[1] Barnard, S.T., "Stochastic Stereo Matching Over Scale," *International Journal of Computer Vision,* 2(4), 1988.

[2] Fua, P.V. and A.J. Hanson, "Extracting Generic Shapes Using Model-Driven Optimization," *Proceedings of the DARPA Image Understanding Workshop,* Cambridge, MA, pp.994–1001, April 1988.

[3] Fischler, M.A., and O. Firschein, *Intelligence: the Eye, the Brain, and the Computer,* Addison-Wesley, Reading, MA, 1987.

[4] Fischler, M.A. and O. Firschein, *Readings in Computer Vision,* Morgan-Kaufmann, Los Altos, CA, 1987.

[5] Leclerc, Y.G., "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision,* 2(4), 1988.

Stochastic Stereo Matching on the Connection Machine

Stephen T. Barnard

Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

February 15, 1989

Abstract

A stochastic approach to stereo matching is presented. A microcanonical version of simulated annealing is used to approximate the ground states of a thermodynamic model system. The potential energy of the system combines two measures of the quality of a dense, two-dimensional disparity map: (1) the photometric error between corresponding points, and (2) the first-order variation (the "flatness") of the map. The method operates over a series of increasingly finer spatial scales. The implementation of this method on the Connection Machine™ is discussed.*

Introduction

Compared to other modes of depth perception, stereo vision seems relatively straightforward. The images received by two eyes are slightly different due to binocular parallax; that is, they exhibit a disparity that varies over the visual field, and that is inversely related to the distance of imaged points from the observer. If we can determine this disparity field we can measure depth and mimic human stereo vision. Few problems in computational vision have been investigated more intensively.

*Support for this work was provided by the Defense Advanced Research Projects Agency under contracts DCA76-85-C-0004 and MDA903-83-C-0084.

We describe an approach to stereo in which the matching problem is posed as computational analogy to a thermodynamic physical system. The state of the system encodes a disparity map that specifies the correspondence between the images. Each such state has an energy that provides a heuristic measure of the "quality" of the correspondence. To solve the stereo matching problem, one looks for the ground state, that is, the state (or states) of lowest energy. This paper is a condensation and revision of an earlier paper [1] and emphasizes some aspects of the implementation on the Connection Machine.^{tm}

Several features of the Connection Machine naturally fit computational vision problems of this type:

- The most important feature is its massive parallelism — up to 64K individual processors, each with 64K bits of memory. Many vision tasks are most naturally expressed as optimization problems on two-dimensional lattices of typically $256 \times 256 = 64K$ pixels.
- The Connection Machine architecture is flexible. It does not restrict the user to a fixed lattice size, or even to one- or two-dimensional lattices. In general, n-dimensional lattices are supported, with n a power of 2 and greater than some bound that depends on the system's configuration.
- Another attractive feature of the Connection Machine is its general method of interprocessor communication. Two varieties of message-passing networks are provided: a boolean n-cube router for general communication and a "NEWS" grid for communication between processors arranged in regular lattices.
- Finally, and not to be underestimated, is the excellent user interface of the Connection Machine, including language processors that are straightforward extensions of standard languages[1] and a graphic display system that provides a high-speed "window" into the system's memory.

The Stereo Matching Problem

Most approaches to stereo matching can be divided into three classes: correlation, feature-matching, and lattice models. Correlation, in its basic form, is the most obvious. Intensity patches in one image are matched to patches in the other image with search, typically using a normalized cross-correlation as a measure of similarity or a normalized mean-square-difference as a measure of dissimilarity. Many variations of this basic theme have been explored. The feature-matching approach matches

[1] We use *Lisp, which is an extension of Common Lisp. Parallel versions of Fortran and C are also available.

directly between discrete sets of points — typically, the output of an edge detector, such as zero-crossing contours. Both suffer from similar difficulties:

- The size of the correlation patch or the support of the feature operator affects the likelihood of false matches. A correlation patch must be large enough to contain the information necessary to specify another patch unambiguously or, failing this, some additional means of disambiguating false matches must be used. Similarly, feature operators with small support will detect many features.
- At the same time, the correlation patch or the feature-operator support must be small compared to the variation in the disparity map. If either is too large, the system will be insensitive to significant relief in the scene.
- In typical images much of the area consists of uniform or slowly varying intensity where neither correlation nor feature matching will be effective.

Lattice models pose the stereo matching problem in terms of optimizing a measure that is usually interpreted as the energy of a lattice of interacting elements. To take one example, Julesz proposed a model consisting of two lattices of spring-loaded magnetic dipoles, representing the two images of a random-dot stereogram [2]. The polarity of the dipoles represents whether pixels in the left and right images are black or white. A state of global fusion is achieved in the ground state, with the attraction or repulsion of the dipoles balanced by the forces of the springs.

A Scaled-Lattice Model

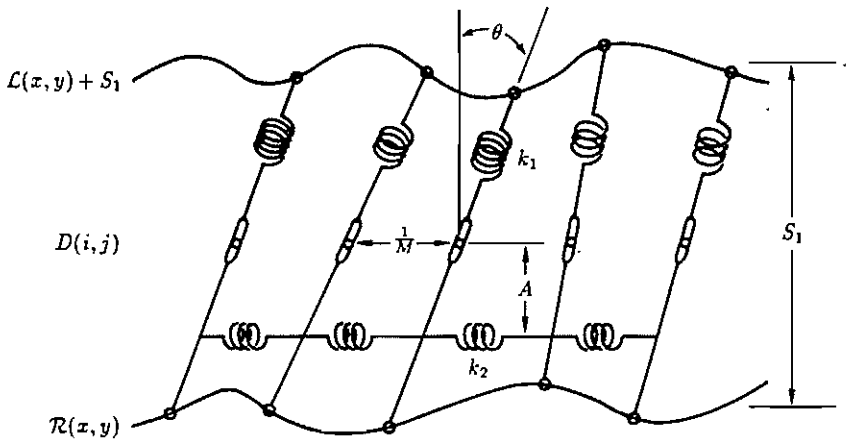
The Stereo Energy Equation

Consider the following equation:

$$\mathcal{E} = \int\int \{(\mathcal{L}(x - \frac{\mathcal{D}}{2}, y) - \mathcal{R}(x + \frac{\mathcal{D}}{2}, y))^2 + \lambda(\nabla\mathcal{D})^2\} dx dy \ , \qquad (1)$$

where $\mathcal{L}$ and $\mathcal{R}$ are piecewise-continuous intensity functions of the left and right visual fields, $\mathcal{D} = \mathcal{D}(x, y)$ is a cyclopean disparity map, and λ is a constant. Each value of $\mathcal{D}$ specifies two corresponding points: $(x - \mathcal{D}/2, y)$ and $(x + \mathcal{D}/2, y)$.

If we assume that $\mathcal{L}$ and $\mathcal{R}$ are commensurate, the first term in the integrand represents the photometric error associated with $\mathcal{D}$. The second term is the first-order variation of $\mathcal{D}$, or a measure $\mathcal{D}$'s "flatness." By minimizing $\mathcal{E}$ with respect to $\mathcal{D}$, therefore, we should find the simplest disparity map (in the sense of flattest) that adequately explains the image data.



vertical springs: spring constant k_1, rest length S_1

horizontal springs: spring constant k_2, rest length $S_2 = \frac{1}{M}$

Figure 1: A spring model

Notice that disparity is a scalar field. Corresponding points may have different x coordinates, but they will always have the same y coordinate. This is a common assumption and involves no loss of generality: if the relative positions and orientations of the two cameras are known, as well as the internal camera parameters, correspondences are restricted to a family of epipolar lines. If the epipolar lines are not horizontal the images can easily be mapped into a normal stereo pair in which they are. We can write the 3D coordinates of the scene in the coordinate frame of the left camera as:

$$\mathrm{p}(x,y) = \frac{B}{\mathcal{D}}(x - \frac{\mathcal{D}}{2}, y, f),$$

where B is the baseline separation and f is the focal length.

Because we will refer to $\mathcal{E}$ as the energy of our system, it is helpful to have a picture of why this interpretation makes sense. One can readily see [1] that $\mathcal{E}$ corresponds to the potential energy of a system of coupled springs illustrated in one dimension in Figure 1.

The model consists of two surfaces, $\mathcal{R}(x,y)$ below and $\mathcal{L}(x,y)+S_1$ above. Midway between these surfaces is a lattice of pivot points, and at each such point is an elastic lever arm, with rest length S_1 and spring constant k_1. The lever arms are free to rotate in the (x,z) plane (i.e., in epipolar planes), while their endpoints are constrained to lie on the two surfaces. The lever arms are connected to their neighbors by other springs with spring constant k_2 that exert torques over moment arm A. The angles of the lever arms represent disparity on an M^2 cyclopean lattice.

It is easy to show that the energy of this system is proportional to $\mathcal{E}$, with

$$\lambda = \left(\frac{k_2}{k_1}\right)\left(\frac{A}{S_1}\right)^2 .$$

Approaches to Minimizing E

Minimizing $\mathcal{E}$ directly is difficult because it is nonlinear. Witkin et al. described a method for optimizing a generalization of Eq. (1) that is essentially a sophisticated form of gradient descent that tracks the solution over increasingly finer scales [3]. The hope is that $\mathcal{E}$ is convex at a coarse scale and that relatively coarse intermediate solutions will place the system in the correct convex region at finer scales. They report that the method is prone to error when it encounters bifurcations in its trajectory. As the scale becomes finer, the system must "choose" which path to follow, and it cannot recover from a mistake because $\mathcal{E}$ may never increase. The solution is therefore critically dependent on initial conditions.

Another approach would be to simulate the dynamics of the spring model. A physical realization of the spring model would be a dynamic system of oscillators that would follow a trajectory through a $2M^2$ dimensional phase space. (Each lever arm has two degrees of freedom: θ and $\dot{\theta}$.) We could flesh out this model by specifying the moments of inertia and damping coefficients of the lever arms. We could also add a periodic forcing function to add energy to the system, balancing the energy dissipated by damping. Having done this, we could could write the differential equations of motion describing the model's deterministic dynamic behavior. In principle, we could trace the trajectory of the system through its phase space, gradually reducing the amplitude of the forcing function while keeping the system in dynamic equilibrium. There is little point in simulating the dynamics in such detail, however, because we know that even low-dimensional forced oscillators have chaotic attractors [4]. The dynamics will be effectively stochastic.

The remainder of this paper describes an alternative and much less expensive approach. Instead of modeling the full dynamics of the system, it models only the thermodynamics. Kinetic energy is modeled as heat.

A Discrete Model

At this point, we will discretize the problem by defining the lattices D, L, and R on $\mathcal{D}$, $\mathcal{L}$, and $\mathcal{R}$. D now has integer values and is interpreted as:

$$L_{i-\left\lfloor \frac{D_{i,j}}{2} \right\rfloor, j} \text{ corresponds to } R_{i+\left\lceil \frac{D_{i,j}}{2} \right\rceil, j} .$$

Eq. 1 becomes

$$E = \sum_{i,j} \{ [L_{i-\left\lfloor \frac{D_{i,j}}{2} \right\rfloor, j} - R_{i+\left\lceil \frac{D_{i,j}}{2} \right\rceil, j}]^2 + \lambda [\Delta D_{i,j}]^2 \} \tag{2}$$

where

$$\Delta D_{i,j} = \sqrt{\sum_{k,l \in \mathcal{N}_{i,j}} (D_{i,j} - D_{k,l})^2} \, .$$

$\mathcal{N}_{i,j}$ denotes the four nearest neighbors of (i,j). In terms of the spring model, the ends of the lever arms are now constrained to lie on a finite number of positions on the two surfaces.

Scaling

Disparity scales linearly with the size of the image. This suggests that a stereo matching system can begin its search at a coarse scale, find an approximate result, use this result to initialize its search at a finer scale, and so on. This has two benefits: it improves the efficiency of search by allowing the system to work initially in smaller phase spaces, and it reduces the false target problem by using a range of spatial scales. Large features are first detected at coarse scales, and their locations in finer scales are constrained.

We can construct a sequence of n lattices, $\{D^k\}$ for $k = 0, n-1$, which represent disparity maps of increasing precision, defined by the following rule:

$$D^k{}_{i,j} \Rightarrow L^k{}_{i-2^{n-k-1}\left\lfloor \frac{D_{i,j}}{2} \right\rfloor, j} \text{ corresponds to } R^k{}_{i+2^{n-k-1}\left\lceil \frac{D_{i,j}}{2} \right\rceil, j} \, . \tag{3}$$

Suppose the maximum range of disparity between a pair of images is $[-2^{n-1}+1, 2^{n-1}]$. The D^0 that matches these images must be binary, and it should be relatively easy to find the best D^0 because the phase space is relatively small. We can then use $2D^k$ as the initial condition of a search for D^{k+1}, until finally D^{n-1} will match the images with single-pixel precision.

When using this scaling method it is necessary to filter L and R to avoid aliasing. We use a difference-of-Gaussians (DOG) approximation to the Laplacian of a Gaussian. This transform can be computed efficiently by recursively applying a small generating kernel [5] to create a low-pass Gaussian sequence, and then differencing successive low-pass images to construct the bandpass DOG sequence. Low-pass filtering alone is adequate to avoid aliasing, but the bandpass filtering is useful for eliminating low-frequency error.

Stochastic Optimization

Standard (Canonical) Annealing

Simulated annealing is a fairly new technique for solving combinatorial optimization problems. The next section presents a new variety of simulated annealing (called microcanonical annealing) that has several advantages for computer implementation. In this section the basic principles of the standard form of simulated annealing are described to set a context for the introduction of microcanonical annealing.

The most fundamental result of statistical physics is the Boltzmann (or Gibbs) distribution

$$P_i = \frac{\exp(-E_i/kT)}{\sum_\nu \exp(-E_\nu/kT)} ,$$

which gives the probability of finding a system in state i with energy E_i, assuming that the system is in equilibrium with a large heat bath at temperature kT. (The constant k (Boltzmann's constant) converts temperature to units of energy. In the following discussion we will assume that $T = kT$.) The normalizing quantity in the denominator, called the partition function, is a sum over all accessible states.

Physicists would like to be able to calculate macroscopic equlibrium properties of model systems. In 1953 Metropolis et al. [6] described a Monte Carlo algorithm that generates a sequence of states that converges to the Boltzmann distribution in the limit. This method, which simulates the effect of allowing the system to interact with a much larger heat bath, samples what is called the canonical ensemble. Macroscopic parameters can then be calculated without knowledge of the partition function by averaging over long sequences.

The Metropolis algorithm begins in an arbitrary state and then successively generates candidate state transitions ($\nu \rightarrow \nu'$) at random. A transition is accepted with the following probability:

$$Pr(\nu \rightarrow \nu'|\nu,\nu') = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp(-\Delta E/T) & \text{otherwise} \end{cases} \qquad (4)$$

where $\Delta E = E_{\nu'} - E_\nu$. Asymptotic convergence of the Metropolis algorithm to the Boltzmann distribution is guaranteed if the process for generating candidate state transitions is ergodic.

Kirkpatrick et al. [7] and Černy [8] independently recognized a connection between the Metropolis technique and combinatorial optimization problems. If the energy of a state is considered as an objective function to be minimized, the minimum can be approximated by generating sequences at decreasing temperatures, until finally a ground state, or a state with energy very close to a ground state, is reached at $T = 0$. This is analogous to the physical process of annealing.

There are results showing the existence of annealing schedules (i.e., the rate of decrease of temperature) that guarantee convergence to ground states in finite time [9], but these schedules are too slow for practical use. Faster *ad hoc* schedules have been used in many problems with good average-case performance. While these faster schedules may not find an optimal state, they can converge to states that are very close to optimal.

Microcanonical Annealing

Creutz [10] has described an interesting alternative to the Metropolis algorithm. Instead of simulating the effect of a large heat bath, the Creutz algorithm simulates a thermally isolated system in which energy is conserved. Samples are drawn from the microcanonical ensemble. One can imagine the difference between the Metropolis algorithm and the Creutz algorithm as follows. The Metropolis algorithm generates a "cloud" of states, each with, in general, different energies, which fills a volume of phase space. As temperature decreases this volume contracts to one or more ground states. The Creutz algorithm, by contrast, generates states on a constant-energy surface in a somewhat larger phase space. As energy decreases these surfaces shrink to the same set of ground states.

The simplest way to accomplish this is to augment the system with one additional degree of freedom, called a demon, which carries a variable amount of energy, E_D. This demon holds the kinetic energy of the system and, in effect, replaces the heat bath. The total energy of the system is now

$$\begin{aligned} E_{total} &= E_{potential} + E_{kinetic} \\ &= E + E_D \end{aligned}$$

The demon energy, being kinetic, is constrained to be nonnegative. The algorithm accepts all transitions to lower energy states, adding $-\Delta E$ (the energy given up) to E_D. Transitions to higher energy are accepted only when $\Delta E < E_D$, and the energy gained is taken away from E_D. Total energy remains constant.

Microcanonical annealing simply replaces the Metropolis algorithm with the Creutz algorithm. Instead of explicitly reducing temperature, the microcanonical annealing algorithm reduces energy by gradually lowering the value of E_D. Standard arguments can be used to show that at equilibrium E_D assumes a Boltzmann distribution over time [10]:

$$Pr(E_D = E) \propto \exp(-E/T) \ .$$

Temperature therefore emerges as a statistical feature of the system:

$$T = \langle E_D \rangle \ . \tag{5}$$

Microcanonical annealing has several advantages over standard annealing:

- It does not require the evaluation of the transcendental function $\exp(x)$. Of course, in practice this function can be stored in a table, but we would like our algorithm to be suited to fine-grained systems with very limited local memory, like the Connection Machine.
- It is easily implemented with low-precision integer arithmetic — again, a significant advantage for simple hardware implementation.
- In the Metropolis algorithm a state transition is accepted or rejected by comparing $\exp(-\Delta E/T)$ to a random number drawn from a uniform distribution over $[0,1]$, and these numbers should be accurate to high precision. The Creutz algorithm does not require high-quality random numbers.

Implementation Details

The program is implemented in Release 5.0 *LISP and is fully compiled with the *LISP compiler. The Connection Machine at SRI is one-eighth of a full machine (8K vs. 64K processors). We have two Symbolics Lisp Machine front ends.

The input to the program is two images, L and R, and a number n that specifies how many levels of scaling are used. The images are assumed to satisfy the horizontal-epipolar condition and to have dimensions of the form $(2^M, 2^N)$, $M, N > 6$. The Connection Machine is assumed to be booted into the same configuration.[2]

The program then DOG filters L and R to create the sequences of bandpassed images $\{L^k\}$, $\{R^k\}$ for $k = 0, n-1$.

Next, beginning with zero disparity at level 0, the program executes a series of n heating and cooling cycles, using the result at level $k-1$ to initialize D^k, generating a sequence of states:

$$(S_0^0 \cdots S_{l_0}^0) \cdots (S_0^{n-1} \cdots S_{l_{n-1}}^{n-1}).$$

A transition between levels amounts to doubling the disparity values and updating the rule for interpreting D (see Eq. 3).

The value of $T \approx \langle E_D \rangle$ of each state of a typical run is plotted in Figure 2. Notice that this run has four levels of scaling. We raise the temperature to $T = 300$ by successively adding 10 units to each demon on each iteration. The system is allowed to dwell at $T = 300$ for awhile, and is then cooled by removing three units from each demon on each iteration. Except for the last cycle, cooling below about $T = 150$ is wasted effort because the fine structure "frozen in" below this temperature will be destroyed in the next heating cycle.

[2]The Connection Machine allows the user to have more than one configuration simultaneously, but this feature is not required here.

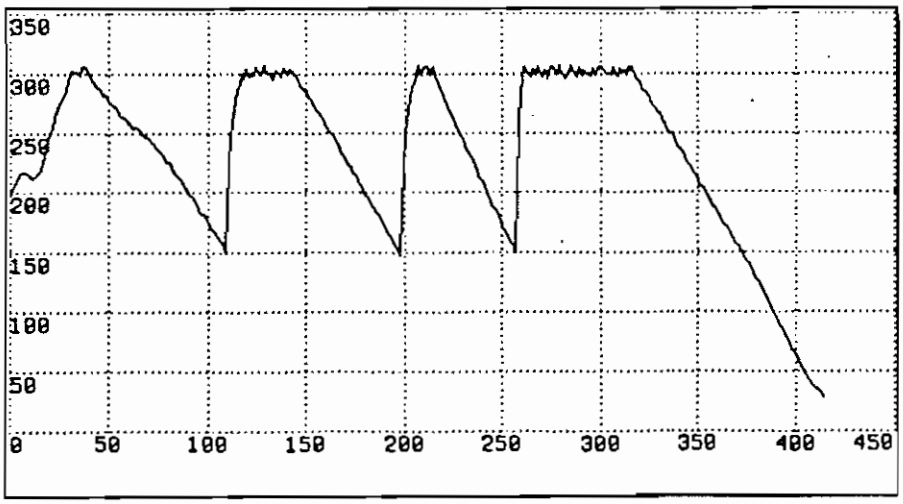


Figure 2: $T = \langle E_D \rangle$

Let r_{eq} be the ratio of the observed average demon energy to the standard deviation of the same observed distribution:

$$r_{eq} = \frac{\langle E_D \rangle}{\sigma(E_D)} \; . \tag{6}$$

At equilibrium E_D will have a Boltzmann distribution, which implies that $r_{eq} = 1$. Figure 3 shows a plot of r_{eq} for the same run as in Figure 2. Note that the plot of r_{eq} indicates that the system moves away from equilibrium during the relatively fast heating cycles, but relaxes quickly back to equilibrium after cooling starts. The system appears to drop away from equilibrium at low temperatures according to the r_{eq} plot, but this effect is actually because there are relatively few energy levels available to the demons near the ground state.

Choosing parameters of this procedure — heating and cooling rates, termination conditions, and so on — remains an art, as in virtually all applications of simulated annealing. The results in Section 0.4 were generated with a common parameter set that was determined empirically from tests on a wide variety of images. A value of $\lambda = 64$ works well for typical images quantized into 8 bits.

As with the Metropolis algorithm, the Creutz algorithm converges to the Boltzmann distribution in the limit for any ergodic process generating candidate state transitions. Of course, different state-transition schemes will affect the rate of convergence. We have found the following simple method to be adequate:

$$P(d \rightarrow d') = \begin{cases} .5 & \text{if } |d - d'| = 1. \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the disparities increase or decrease by one lattice position, or remain unchanged if the transition is rejected, as the system follows a Brownian path on its phase-space surface of constant energy.

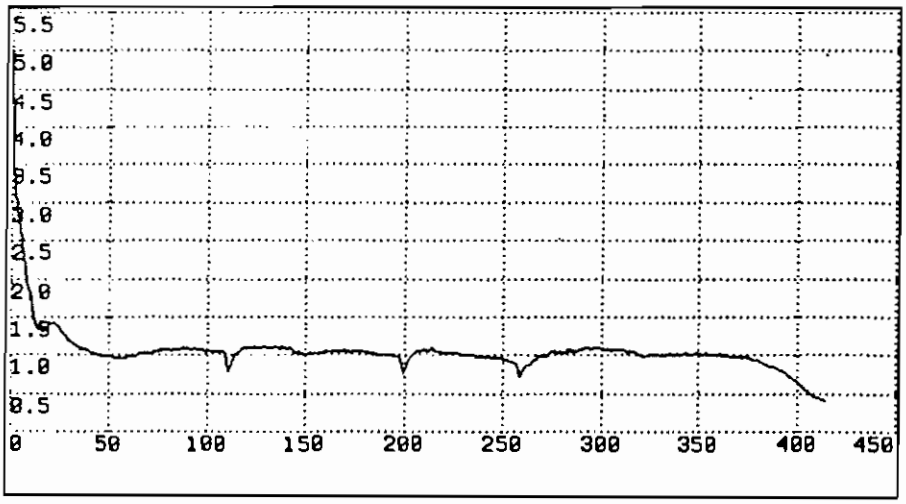


Figure 3: $r_{eq} = \frac{\langle E_D \rangle}{\sigma(E_D)}$

Boundary conditions can be troublesome. Nonzero disparities near the edge of the lattice can match image points off the lattice. When this occurs we assign the photometric term in Eq. 2 a value equal to the current temperature, effectively placing an energy barrier at the boundary.

Annealing in Parallel

The essential inner loop of the algorithm, called a "full-pass", tries exactly one random transition for each lattice site. It is important to realize that we cannot update *all* sites in parallel. One full-pass should leave the total energy $E + E_D$ unchanged, but this is not ensured if two neighbors are updated simultaneously. This presents no problem for four-neighbor interactions: the lattice can be split into two "checkerboard" subsets that are updated sequentially. More complex neighborhoods would require more subsets, reducing parallelism.

The basic version of microcanonical annealing, using only one demon, is not suited to a parallel implementation. Each decision to accept or reject a state transition depends on the value of E_D and, therefore, on the previous decision. Instead, we use a lattice of demons. Temperature is still measured with Eq. 5, but using the distribution of E_D over space rather than time. Statistics can be sampled over both time and space, if desired.

There is a minor complication in using a lattice of demons. The single-demon algorithm visits sites at random and the demon allows energy to be transferred throughout the lattice. Similarly, in the lattice-of-demons algorithm the demons must be mixed throughout the lattice. We use a complete random permutation of the demons after every lattice update, but more local methods are also adequate.

lattice	computer	VP ratio	time	factor-speedup
128^2	SLM	n.a.	19.4 sec.	n.a.
	4K	4	.28 sec	69
	8K	2	.26 sec.	74
256^2	SLM	n.a.	76.7 sec.	n.a.
	4k	8	.54 sec.	142
	8k	4	.36 sec.	213

Table 1: Symbolics 3600 vs. Connection Machine

Experimental Results

This section presents experimental results for two distinct cases: an aerial stereo pair (Figure 4) and a ground-level scene with prominent occlusions (Figure 5).

The figures show the two original images (both are 128 $\times$ 128) and the disparity map at the end of each cooling cycle. Each example required about 4 minutes of processing time.

Table 1 indicates how different Connection Machine configurations compare to a Symbolics 3600 Lisp Machine. The timings are for one full-pass. Note that the efficiency of the Connection Machine depends strongly on the VP ratio, which is the ratio of the number of virtual processors to physical processors. This is because the overhead of the front end is too large to keep up with the Connection Machine at low VP ratios.

Conclusions

The method fits the Connection Machine architecture very well and was quite easy to implement. The processing is still too slow for real-time applications, but is adequate for cartography. By using a new feature of the Connection Machine software that allows the user to define virtual processor sets of different sizes, the implementation will be able to process $1K \times 1K$ images in a reasonable time.

The use of a scale hierarchy dramatically increases the efficiency of the method, especially for large problems such as those illustrated in Figures 4 and 5. An additional benefit of using a scale hierarchy is that the solution is less sensitive to small

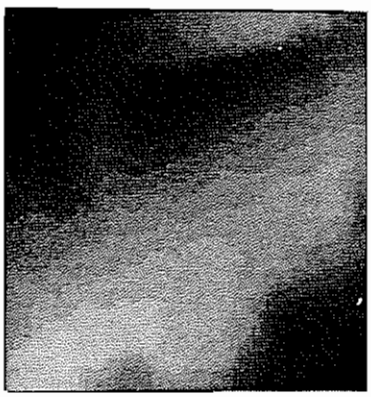
(a) Left image.

(b) Right image.

(c) $k = 0, T = 150$.

(d) $k = 1, T = 150$.

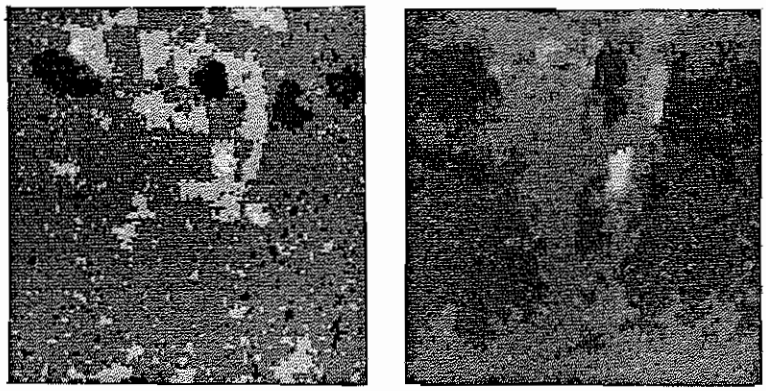
(e) $k = 2, T = 150$.



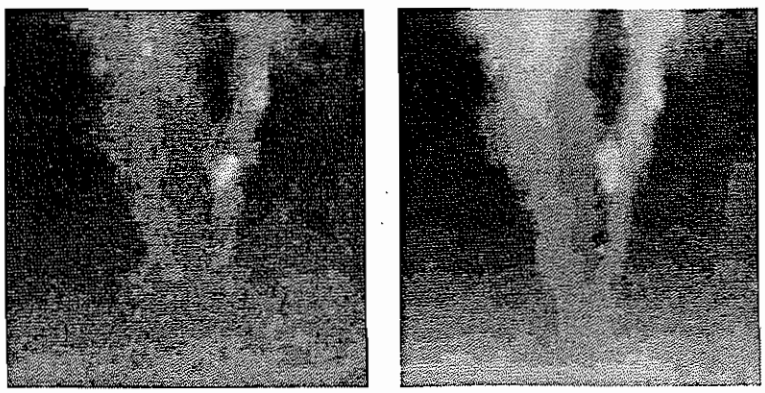
(f) $k = 3, T = 0$.

Figure 4: Aerial stereogram results

(a) Left image. (b) Right image.



(c) $k = 0, T = 150$. (d) $k = 1, T = 150$.



(e) $k = 2, T = 150$. (f) $k = 3, T = 0$.

Figure 5: Ground-level stereogram results

amounts of vertical disparity, which is eliminated at coarser scales. (Uncertainty in the camera model will usually cause some vertical disparity in high-resolution images.) A Gaussian low-pass hierarchy works as well as the Laplacian hierarchy if the images are recorded with equivalent sensors. The benefit of bandpass filtering is to eliminate the low-frequency variation caused by uncalibrated photometry. Annealing provides a way to bridge the gap between scales. The microcanonical annealing algorithm appears to be an improvement over canonical annealing for reasons discussed in the section on that algorithm.

Canonical annealing and "pure" single-demon microcanonical annealing are at opposite ends of a spectrum. In canonical annealing the heat bath is much larger than the model system, and is not represented explicitly. In pure microcanonical annealing the heat bath — that is, the single demon — is much smaller than the system, and it is represented explicitly. The lattice-of-demons algorithm is midway between these extremes, with the heat bath and the model system having comparable sizes. In a sense, this is a classical space/time tradeoff. By representing the heat bath explicitly we can avoid the evaluation of complicated functions.

References

[1] Barnard, S.T., "Stochastic Stereo Matching Over Scale," *International Journal of Computer Vision,* Vol. 2(4), 1988.

[2] Julesz, B., *Foundations of Cyclopean Perception,* University of Chicago Press, Chicago, IL, 1971.

[3] Witkin, A., D. Terzopoulos, and M. Kass, "Signal Matching Through Scale Space," *International Journal of Computer Vision,* Vol. 1, pp.133–144, 1987.

[4] Walker, G.H., and J. Ford, "Amplitude Instability and Ergodic Behavior for Conservative Nonlinear Oscillator Systems," *Physics Review,* Vol. 188, pp.416–432, 1969.

[5] Burt, P., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications,* Vol. COM-31, pp.532–540, 1983.

[6] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics,* Vol. 21, pp.1087–1092, 1953.

[7] Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,*, Vol. 220, pp.671–680, 1983.

[8] Černy, V., "Thermodynamical Approach to the Traveling Salesman Problem: An efficient Simulation Algorithm," *Journal of Optimization Theory and Applications,* Vol. 45, pp.41–51, 1985.

[9] Geman, S., and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 6, pp.721–741, 1984.

[10] Creutz, M., "Microcanonical Monte Carlo Simulation," *Physical Review Letters,* Vol. 50, pp.1411–1414, 1983.

Object Delineation as an Optimization Problem, a Connection Machine Implementation

Pascal Fua

Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

Abstract

In real-world imagery, segmentation methods that rely on local image properties often fail to extract semantically meaningful features. We propose an objective function that exploits all the available photometric information. We take advantage of parallelism to effectively compute and optimize this objective function in order to find object outlines. We present our Connection Machine™ implementation and show how this technique can be used to delineate complex objects in aerial imagery and determine their elevation when using stereo pairs of images.

Introduction

In real-world imagery, object boundaries cannot be detected solely on the basis of their edge photometry because of the presence of noise and photometric anomalies. Thus, methods for delineating objects based on purely local statistical criteria are bound to make mistakes; no single parameter setting can be effective over different areas of a single image, much less for multiple images.

We address this problem by introducing "score optimizing curves" that describe objects as smooth or polygonal curves that enclose an area in the image. A global

*This research was supported in part by the Defense Advanced Research Projects Agency under contracts DCA76-85-C-0004 and MDA903-83-C-0084.

score is formed from these curves utilizing both edge information on the curve itself and the photometric information in the entire delineated area. A parallel optimization procedure deforms the curves to maximize the score and conform to object outlines.

Parallelism provides the computational power for performing the optimization in real time: at every iteration of the optimization procedure, one must recompute the photometric characteristics of the curve and its enclosed area. While this procedure could be implemented on a serial machine, the computational burden increases with the size of the enclosed area, making the optimization unacceptably slow for large objects.

Such "score optimizing curves" were originated by Terzopoulos, Kass, and Witkin as "snakes" [8, 14]. In their approach, boundaries are described as polygonal curves with a score that includes geometrical constraints and a measure of edge strength. "Snakes" do not take into account any photometric evidence outside the edge; they yield good results only if the initial position of the curve is close enough to the boundary of the object to be influenced by its edges. Because we also use *interior area* information, our curves can easily grow or shrink if the initial position is very inaccurate. By integrating more information, our algorithm also becomes more stable and less sensitive to photometric anomalies. Furthermore, in this framework, we can also take advantage of depth information and determine the elevation of an object when working with stereo pairs of images.

In this paper, we first introduce our objective function. We then describe a parallel implementation of the optimization procedure on a Connection Machine^{tm} and show how this technique can be used to delineate complex objects in aerial imagery and determine their elevation.

Objective Function

Our goal is to extract objects that conform to a particular photometric model. To discriminate among competing hypotheses, we need an objective function that measures the goodness of fit to feature models including such characteristics as area, edge, and stereo photometry, as well as shape or semantic relationships.

Because we must be able to combine corresponding measures for these models in a commensurate fashion, we choose an information-theoretic approach that enforces compatibility of the various measures. For each photometric model, we compute what we call the *effectiveness* F of the model that we define as *the difference between the number of bits needed to encode the photometry of a scene patch without the model versus with the model.* F is largest for patches that conform well to the model and can therefore be described effectively in terms of it; F measures the goodness of fit to the model. For a theoretical justification of this approach, we refer the reader to the

Minimal Description Principle introduced by Rissanen [10, 11]. Our method shares many basic concepts with the information-theory approach to segmentation described by Leclerc [9] in these proceedings; however, because our goal is to extract objects of interest rather than to segment the whole scene, many additional issues arise.

Two of the main characteristics of an object in an image are its interior texture and its contrast with the background, which produces edges. Here we explore simple models for the textured area and for the edges of an object that have proven useful in analyzing aerial imagery. The photometric evidence relevant to the edges comes from background pixels that are independent of pixels interior to regions. Therefore, these two measures are independent and we take the complete objective function F to be the sum of area and edge components, $F = F_A + F_E$. When analyzing stereo pairs of images, we also use a stereoscopic model and compute the elevation parameters of an object in the scene by optimizing the corresponding stereo effectiveness F_S.

A more robust objective function would also include a term that measures the geometric quality of a given curve and its conformity to a geometric model [3]. We have not yet incorporated such a measure in our Connection Machine implementation, and a complete discussion of a geometric term is therefore beyond the scope of this paper. However, optimizing F by itself is impractical because the "score optimizing curve" would lose its shape during the optimization. As suggested by Terzopoulos, Kass, et al. [14], we address this problem by introducing a deformation energy D that increases when the curve becomes irregular and optimizing $F - D$ instead of F; this point is discussed in detail in the implementation section.

Essential Parameters of the Objective Function

We introduce two fundamental parameters, the *scale* and the *shape coefficient*:

- **Scale.** The scale is interpretable as the unavoidable dimensional factor that converts dimensional quantities like area or length into dimensionless probabilities. Area units are thus scaled down by two powers of the dimensional unit, while boundary lengths are scaled down by a single power. The scale parameter thus controls whether or not area signature dominates edge signature.

- **Shape Coefficient.** Because we introduce the deformation energy D in our optimization, we must weigh its contribution using a shape coefficient. In our implementation

  $$D = \lambda L^2$$

  where L is the squared length of the boundary of the patch and λ the shape coefficient. D is a smoothing term required to enforce regularity of the boundaries because F is a highly nonconvex function that would be difficult to optimize by itself; λ controls the amount of smoothing.

We know of no *a priori* way to determine the scale and shape coefficient, because they characterize the fundamental balance of influences that must be specified for each application. Nevertheless, our approach provides a clear way to justify and understand the essential roles of these two parameters in feature extraction.

Area Model for Homogeneous Regions

We model a homogeneous region with area A, such as a building roof, as a planar intensity surface with a Gaussian distribution of deviations from the plane, plus anomalous pixels whose values lie outside the peak of the distribution.

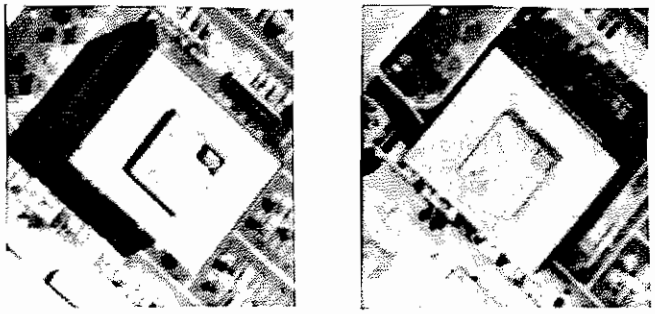


Figure 1: A stereo pair of images containing a large building

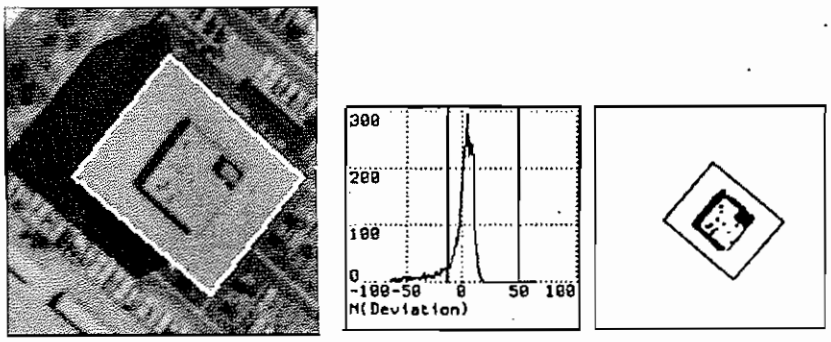
Figure 1 shows a stereo pair of images, Figure 2 a the outline of the main rooftop in the left image, and Figure 2 b the corresponding histogram of deviations from the planar fit to the intensity surface along with the left and right bounds of the main Gaussian peak. In Figure 2 c, the solid white area indicates the location of the pixels within the peak. Black areas within the outline lie outside the peak and are considered anomalous.

In an 8-bit image, it would take $8A$ bits to encode the pixel values if we did not take advantage of dependencies among pixels. Similarly it would take $k_A A$ bits to encode the same information using our region model, where

$$k_A A = n(\log \sigma + c) + 8\overline{n} - \left[n \log \frac{n}{A} + \overline{n} \log \frac{\overline{n}}{A} \right]. \qquad (1)$$

Here σ is the variance of the Gaussian distribution, n is the number of pixels in the Gaussian, and $\overline{n} = A - n$, and $c = \frac{1}{2}\log(2\pi e)$*. Note that in the computation of the

*All logarithms in this paper are base 2 logarithms.



(a) (b) (c)

Figure 2: (a) Outline of the main rooftop in the left image of Figure 1 (b) Histogram of deviations from the planar fit with left and right bounds of the Gaussian peak. (c) The solid black areas within the contour indicate the location of the pixels that do not belong to the main Gaussian peak and are considered anomalous.

encoding cost, we have not included the cost of encoding the internal parameters of the model, such as the slopes and intercept of the plane. It can be shown [10, 13] that these costs are proportional to the logarithm of the area A and are therefore very small compared to $k_A A$.

We weight all areas and lengths using the *scale parameter* s, so that the area effectiveness becomes

$$\begin{aligned} F_A &= \text{bits(area without model)} - \text{bits(area with model)} \\ &= (8 - k_A)\frac{A}{s^2}. \end{aligned} \tag{2}$$

Effect of anomaly discounting. In the left-hand graphs in Figure 3, we plot the area effectiveness F_A as a function of the radius of a square-shaped patch at the center of the images shown in the left column: a good but noisy synthetic image of a square, the same image with edge jitter, and with gross area anomalies. When we compare the results obtained *after discounting anomalies* (solid lines) with those found without anomaly discounting (dotted lines), we see that anomaly discounting can easily be *entirely responsible* for generating the local extrema (i.e., the desired shape) perceived by human observers. This is potentially a critical factor in the practical application of this approach because, as we see in Figure 2, real images nearly always have significant anomalous components.

Parallel computation of the score. The score can be efficiently evaluated on a Connection Machine because the computation only involves fitting a plane over a patch, computing the deviations histogram, and finding for the Gaussian peak the left and right bounds that yield the best value of F_A. The planar fit requires a small number of parallel summations, and the histogram can be computed in one parallel operation. All possible choices of the left and right bounds of the peak are then evaluated simultaneously, and only the best are retained.

Edge Model

We adopt the definition [2, 7, 12] of edge pixels as maxima of the local image derivative. To enforce this criterion in our information theoretic framework, we propose the following scheme.

We take the edge gradient to be $g = (\partial I/\partial x)^2 + (\partial I/\partial y)^2$ where I is the image intensity. Assuming that g ranges between 0 and M, it would take $logM$ bits per pixel to encode gradient intensities of boundary pixels in the absence of a model. The gradient of boundary pixels is expected to be higher than that of other pixels; we model this fact by describing the edge strength in terms of a vocabulary that favors high gradients. We assume that a pixel with gradient g can be described using $-\log(g/c)$ bits where $c = M^2/2$ is a normalizing constant (g/c is a probability density that must sum to 1 over all possible values of g).

We then weight all lengths by the scale factor s and estimate the edge effectiveness to be, for a boundary of length L,

$$\begin{aligned} F_E &= \text{bits(edge without model)} - \text{bits(edge with model)} \\ &= \frac{L}{s}\log M + \frac{1}{s}\sum \log \frac{g}{c} \\ &= \frac{1}{s}\sum \log \frac{g}{\gamma} \end{aligned} \tag{3}$$

where $\gamma = M/2$ and $\sum$ represents a summation over the boundary pixels.

In practice γ is computed as a percentage of the edge strength and treated as a threshold value for the edge strength under which $\log(g/\gamma)$ is taken to be 0. It can be shown [4] that all the points along a curve that maximizes F_E are maxima of the edge gradient in the direction normal to the curve and therefore satisfy our definition of an edge pixel.

The right-hand graphs in Figure 3 show the edge effectiveness of the boundaries of the square patches discussed in the previous section, as a function of their radius. In Figure 4, we plot the area and edge scores (with anomaly discounting) as a function of size when square patches (solid lines) are compared to circular ones (dotted lines), as applied to the images of Figure 3. In the case of the perfect square, the edge

score clearly provides excellent discriminating power. However, in the case of the square with edge jitter, the optimum of the edge effectiveness is much less distinct; the combination of edge effectiveness and area effectiveness has more discriminating power than either alone. Also note that the differences in effectiveness between the squares and circles are much less marked in the noisy image than in the noise-free one. This is an intuitively satisfactory behavior because the square shape is much less perceptible in the noisy image.

Parallel computation of the score. The image gradients can be precomputed using Gaussian convolution operators. The computation of the score then reduces to a global summation of the gradient intensities over the boundary pixels, which can be achieved in one parallel step.

Stereography

The simplest stereo model assumes that corresponding pixels have the same grey-levels in both images [1]. In practice, one finds deviations from this model that we encode again as a Gaussian distribution, excluding anomalies arising from such causes as occluding structures.

As in the area-encoding case, we can now determine the number of bits required to encode the area in the second image by histogramming the deviations of the intensities from their predicted values. We also want to take into account the edge quality of the contour in the second image and its edge effectiveness.

We therefore take the stereographic effectiveness term F_S to be the sum of an edge and area term:

$$\begin{aligned} F_S &= F_{AS} + F_{ES} \\ F_{AS} &= (8 - k_2)\frac{A_2}{s^2} \\ F_{ES} &= \frac{1}{s}\sum \log \frac{g}{\gamma} \end{aligned} \tag{4}$$

where A_2 is the area of the projected patch in the second image, k_2 is the average number of bits/pixel needed to encode the deviation histogram, and g the edge gradient in the second image.

We can use the effectiveness measure Eq. 5 to optimize the elevation parameters of a two-dimensional delineation found in the first image. The search space is extremely constrained because the projected shape is known and the only degree of freedom is epipolar motion in the second image.

Let us consider the stereo pair of images shown in Figure 1 and the rooftop outlined in Figure 5a. Assuming that it is horizontal, we plot in Figure 5b the value of F_S

as a function of the assumed disparity between the outline in the left image and the outline in the right image. We note that F_S presents a sharp peak for the correct match shown in Figure 5c.

Implementation and Applications

Deformable Models in Two Dimensions

To find local maxima of the objective function $F - D$, where $F = F_E + F_A$ and $D = \lambda L^2$ is the deformation energy introduced in Section , we describe object contours as deformable closed curves defined by an ordered list of contiguous points C represented by the vector X of their integer x coordinates and the vector Y of their y coordinates. During each iteration of the optimization procedure described below, X and Y are updated. C is then recomputed by drawing scan lines between points that are not contiguous anymore and merging points that have identical coordinates, thereby generating new vectors X and Y. The edge effectiveness F_E is computed using those new boundary pixels and the area effectiveness F_A of the pixels enclosed by the boundary but not belonging to it. In this way the contour can shrink or expand as required to optimize the objective function.

At every iteration, we compute the derivative of the F with respect to deformations of the contour C:

$$\begin{aligned} \frac{\partial F}{\partial X} &= \frac{\partial F_A}{\partial X} + \frac{\partial F_E}{\partial X} \\ \frac{\partial F}{\partial Y} &= \frac{\partial F_A}{\partial Y} + \frac{\partial F_E}{\partial Y} \end{aligned} .$$

In the appendix we derive expressions for these derivatives and show that they can be easily evaluated on a Connection Machine.

To perform the optimization we could use a simple gradient descent technique, but it would be extremely slow for curves with a large number of points. Instead, we modify the standard gradient procedure in two ways:

1. **Treat C as a physical system.** As in the work by Terzoupolus [14], we consider C as a "snake" that is, a physical curve defined by the vector (X, Y), embedded in a medium of viscosity $\alpha = 1/\lambda$, and moving under the influence of the potential $V = L^2 - \alpha F$. L^2, the square length of the boundary, can be computed as:

$$L^2 = \frac{1}{2} XKX + \frac{1}{2} YKY \tag{5}$$

where K is the tridiagonal matrix with coefficients $-1, 2, -1$. At every iteration of the optimization, we then solve the equation of dynamics:

$$\frac{\partial V}{\partial C} + \alpha \frac{dC}{dt} = 0 \tag{6}$$

where $\partial V/\partial C$ is the vector $(\partial V/\partial X, \partial V/\partial Y)$. Because the deformation energy L^2 in Eq. 5 is quadratic, its derivatives with respect to X and Y are linear. Thus, each iteration of the optimization amounts to solving the two linear equations:

$$\begin{aligned} KX_t + \alpha(X_t - X_{t-1}) &= \alpha \left.\frac{\partial F}{\partial X}\right|_{C_{t-1}} \\ KY_t + \alpha(Y_t - Y_{t-1}) &= \alpha \left.\frac{\partial F}{\partial Y}\right|_{C_{t-1}} . \end{aligned} \tag{7}$$

Letting $M = (I + \frac{1}{\alpha}K)^{-1}$, Eq. 7 can be rewritten as:

$$\begin{aligned} X_t &= M(X_{t-1} + \left.\frac{\partial F}{\partial X}\right|_{C_{t-1}}) \\ Y_t &= M(Y_{t-1} + \left.\frac{\partial F}{\partial Y}\right|_{C_{t-1}}) . \end{aligned} \tag{8}$$

For α large enough (typically $\alpha > .01$), the matrix M can be approximated with excellent accuracy by an n-diagonal matrix. We can therefore solve Eq. 8 simultaneously for X and Y by convolving the right-hand terms $X + \partial F/\partial X$ and $Y + \partial F/\partial Y$ with the appropriate mask. In this formulation, the value of α determines the width of the mask and how much X and Y are smoothed – the smaller α, the more smoothing.

It is worth noting that approximately the same result can be achieved by a faster although slightly less accurate procedure. Instead of solving the equations of the dynamics, we can increment X and Y by $\partial F/\partial X$ and $\partial F/\partial Y$ as in a standard gradient procedure and then recursively smooth the resulting coordinate vectors using the mask $[.25, .5, .25]$. This procedure is fast because it can be implemented using only integer additions and left shifts but no floating point operations or multiplications. In practice, the results produced by these two procedures are almost indistinguishable. The results presented in this paper have been generated using recursive smoothing; at each iteration, the X and Y vectors are convolved 10 times with the mask $[.25, .5, .25]$ except in the example shown later in Figure 7d.

2. **Normalize the derivatives of the score.** The magnitude of the derivatives is not related to the current distance of the contour to its optimal location. Therefore, for every iteration, we pick a step size and retain only the sign of the

derivatives that indicates in which direction the contour should move, resulting in a string FX of -1,0 and 1 for the X coordinates and a string FY for the Y coordinates. We then normalize the string so that $(\|FX\|^2 + \|FY\|^2)/n = \delta^2$, where n is the number of boundary points and δ the current step size, and replace $\partial F/\partial X$ and $\partial F/\partial Y$ by FX and FY in Eq. 8. This ensures that the displacement of each point is on the average of magnitude δ.

Because of the presence of the linear terms in the dynamics equation (Eq. 6), deformations are propagated along the whole curve at every iteration, making this procedure considerably faster than ordinary gradient descent.

Because the objective function is highly nonconvex, after each iteration we recompute the score and verify that it has increased. If, instead, it has decreased, the curve is reset to its previous position and the step size reduced.

The optimization proceeds until the curve stabilizes. For example, going from the initial estimates of the closed curve shown in Figure 6a to the final result shown in Figure 6d took only 10 iterations. Figure 6b and 6c show the position of the curve after three and five iterations respectively.

We now turn to the aerial image in Figure 7a. The four initial contours shown in Figure 7b yield, after optimization, the final outlines of Figure 7c. Note that the corners of the house are slightly rounded due to the presence of the smoothing term. To delineate the house more accurately, we can reoptimize the corresponding curve using less smoothing, generating the result shown in Figure 7d.

Timing considerations. As long as there are fewer points in the "score optimizing curve" C than there are Connection Machine processors, most of the computation can be performed with a virtual processor ratio of 1, with the possible exception of the planar fit and the computation of the deviation histogram required for estimating F_A, which must refer directly to the image. In the following table, we indicate the average times required to perform the various operations when dealing with either a a 64×64 image or a 128×128 image on the 8K machine we are using.

Time in seconds to:	64×64	128×128
Compute F_A	.165	.185
Compute F_E	.011	.011
Compute $\partial F/\partial X$ and $\partial F/\partial Y$	.055	.055
Update X and Y	.180	.195
Perform a complete iteration	.42	.45

If we were using a full 64K machine, the times would be the same for 256×256 and 512×512 images respectively. Note that these times are independent of the

length of the contour or its interior area. If this algorithm were implemented on a serial computer, the time required to compute F_E would grow as the length of the boundary, and the time required to compute F_A would grow as the area, which would slow down the algorithm unacceptably for any large object.

Polygonal Models in Two and Three Dimensions

The "score optimizing curves" described in the previous section behave like rubber bands that attempt to shrink-wrap the contours of an object and yield a smoothed outline. When attempting to extract polygonal objects, we can explicitly include a polygonal constraint by fitting line segments to the curve after each iteration of the optimization procedure.

We now consider the left image of the stereo pair shown in Figure 1. In Figure 8a we show three initial polygonal contours, and in Figure 8b the result of the optimization assuming that the number of vertices in the contours does not change. In the presence of corners, the polygonal constraint yields better results, provided that the location of the polygon vertices can be computed. In practice, this can be achieved by first performing the optimization with a simple rubber band, finding the high curvature points, and using these as candidate vertices for a polygonal "score optimizing curve".

After the contour outlines shown in Figure 8b are found, their elevation can be determined by optimizing the value of the stereo effectiveness from Eq. 5. Assuming that the rooftops are planes, the matching contours in the right image are shown in Figure 8c. These contours and their elevation can then be fed to a system such as the SRI cartographic modeling system [5, 6] to generate synthetic three-dimensional views of the scene.

Conclusion

We have presented, for automatically outlining object boundaries, a technique that integrates area, edge, and stereographic information with geometric models, given a very rough initial estimate of the boundary. The constraints are incorporated by defining, for curves, an objective function that is maximal when the models are satisfied exactly. The initial estimate is used as the starting point for finding a local maximum of this objective function by embedding the initial curve in a viscous medium and solving the equations of dynamics.

The strength of this "score maximizing curve" approach is that all the available photometric information is taken into account simultaneously with geometric constraints.

Parallelism is essential for a successful implementation of this technique because it provides the computational power required to perform the optimization in real time. We plan to apply this technique to investigate more sophisticated constraints, including more sophisticated geometric models than the one described in this paper, and to better understand their relevance to the feature extraction problem. Our Connection Machine implementation and our optimization scheme will allow us to quickly experiment with such constraints on numerous examples and decide their value. Such an investigation would be impractical without the possibility of performing such experiments rapidly.

This technique can also be used for semiautomated data acquisition: a photo-interpreter provides a very rough estimate of the location of an object and lets the computer determine the object's precise outline and elevation. In future work, therefore, another goal will be to provide the user with means of interactively guiding the optimization when necessary and to introduce geometric constraints that objects of interest must satisfy.

Appendix: Derivatives of the Effectiveness

Derivatives of the Area Term

To estimate the derivatives of F_A, we first compute the contribution dF_A of every point (x, y) in the image when added to the patch defined by C. As shown in Eq.2

$$\begin{aligned} F_A &= (8 - k_A)\frac{A}{s^2} \quad \text{where :} \\ k_A &= n(\log\sigma + c) + 8\overline{n} + \left[n\log\frac{n}{A} + \overline{n}\log\frac{\overline{n}}{A}\right] \\ c &= \frac{1}{2}\log(2\pi e)\,, \end{aligned}$$

which we can rewrite as:

$$k_A = n(c_1 - \frac{\log v}{2}) + n\log n + \overline{n}\log\overline{n} - A\log A,$$

where $c_1 = 8.0 - c$ and $v = \sigma^2$. To evaluate the contribution of an individual pixel we must distinguish two different cases:

1. The pixel belongs to the main gaussian peak if its deviation from the planar fit d is between the left and right bounds defined in section . In that case, n and A must be incremented by 1 while the the overall variance v is modified by

$dv \approx (d^2 - v)/n$. Therefore dF_A can be computed as follows:

$$\begin{aligned} dF_A &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2} n \frac{dv}{v} + \log n - \log A \\ &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2}(\frac{d^2}{v} - 1) + \log n - \log A \end{aligned}$$

where $c_2 = \log_e 2$.

2. The pixel does not belong to the main peak, its contribution to $\overline{n}$ and dF_A can be taken as:

$$dF_A = \log \overline{n} - \log A.$$

Having computed dF_A, we can now estimate $\partial F_A/\partial X$ using finite differences. Let us consider a boundary point $P = (x, y)$. Our implementation assumes that the boundary points themselves do not belong to the patch. There are four possible patterns for the 3×1 horizontal neighborhood centered around P:

a:	1	×	0
b:	0	×	1
c:	1	×	1
d:	0	×	0

where 0 represents a point that does not belong to the patch and 1 represents a point that does.

- Case a: If P moves to the right, the center point is added to the patch and F_A becomes $F_A + dF_A(x, y)$; conversely if P moves to the left, the left point is removed from the patch and the F_A becomes $F_A - dF_A(x-1, y)$. $\partial F_A/\partial x$ is therefore estimated to be:

$$\frac{\partial F_A}{\partial x} = + \frac{dF_A(x, y) + dF_A(x-1, y)}{2}.$$

- Case b: Similarly,

$$\frac{\partial F_A}{\partial x} = - \frac{dF_A(x, y) + dF_A(x+1, y)}{2}.$$

- Case c and d: The boundary is locally horizontal,

$$\frac{\partial F_A}{\partial x} = 0.$$

$\partial F_A/\partial X$ is the vector of the $\partial F_A/\partial x$ for all the points in C. $\partial F_A/\partial Y$ is computed similarly by replacing horizontal neighborhoods by vertical ones. Note that dF_A can be computed on a pixel-per-pixel basis and therefore in parallel for all pixels in the image. The computation of $\partial F_A/\partial X$ and $\partial F_A/\partial Y$ involves only communication with nearest horizontal and vertical neighbors — operations that are very fast on a Connection Machine^{tm}.

Derivatives of the Edge Term

We have seen in Eq. 3 that F_E is computed as

$$F_E = \frac{1}{s} \sum_{C(x,y)} \log \frac{g(x,y)}{\gamma}.$$

In practice we precompute, once and for all, the quantity Γ defined by

$$\Gamma(x,y) = \begin{cases} \log(g(x,y)/\gamma) & \text{if } s > \gamma \\ 0 & \text{otherwise} \end{cases}.$$

We also precompute the derivative of Γ, $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$. At each iteration, $\partial F_E/\partial X$ and $\partial F_E/\partial Y$ are simply the vectors whose components are the values of $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$ at the current boundary points.

References

[1] Barnard, S.T., "Stochastic Stereo Matching Over Scale," *Proceedings of the DARPA Image Understanding Workshop,* Boston, MA, pp.769–778, April 1988.

[2] Canny, J., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 8(6), pp.679–698, 1986.

[3] Fua, P.V., and A.J. Hanson, "Extracting Generic Shapes Using Model-Driven Optimization," *Proceedings of the DARPA Image Understanding Workshop,* Boston, MA, pp.994–1004, April 1988.

[4] Fua, P.V., and Y.G. Leclerc, "Model Driven Edge Detection," to be published in the *Journal of Machine Vision and Applications,* 1989.

[5] Hanson, A.J., A.P. Pentland, and L.H. Quam, "Design of a Prototype Interactive Cartographic Display and Analysis Environment," *Proceedings of the Image Understanding Workshop,* pp.475–482, February 1987.

[6] Hanson, A.J., and L. Quam, "Overview of the SRI Cartographic Modeling Environment," in *Proceedings of the Image Understanding Workshop,* Boston, MA, pp.576–582, April 1988.

[7] Haralick, R.M., "Digital Step Edges from Zero Crossings of Second Directional Derivatives," *IEEE Transactions on Pattern Analysis and Machine Vision,* Vol. 6(1), pp.58–68, 1984.

[8] Kass, M., A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision,* Vol. 1(4), pp.321–331, 1988.

[9] Leclerc, Y.G., "Segmentation Via Minimal-Length Encoding on the Connection Machine," *Proceedings. of the Fourth International Conference on Supercomputing,* Santa Clara, CA, April–May 1989.

[10] Rissanen, J., "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics* Vol. 5, pp.416–431, 1983.

[11] Rissanen, J., "Minimum-Description-Length Principle," in *Encyclopedia of Statistical Sciences*, Vol. 5, pp.523-527, 1987.

[12] Rosenfeld, A., "A Nonlinear Edge Detection Technique," *Proceedings of the IEEE,* Vol. 58, pp.814–816, 1970.

[13] Schwarz, G., "Estimating the Dimension of a Model," *The Annals of Statistics,* Vol. 6, pp.461–464, 1978.

[14] Terzopoulos, D., "On Matching Deformable Models to Images," *Topical Meeting on Machine Vision,* Technical Digest Series, Optical Society of America, Washington, DC, Vol. 12, pp.160–167, 1987.

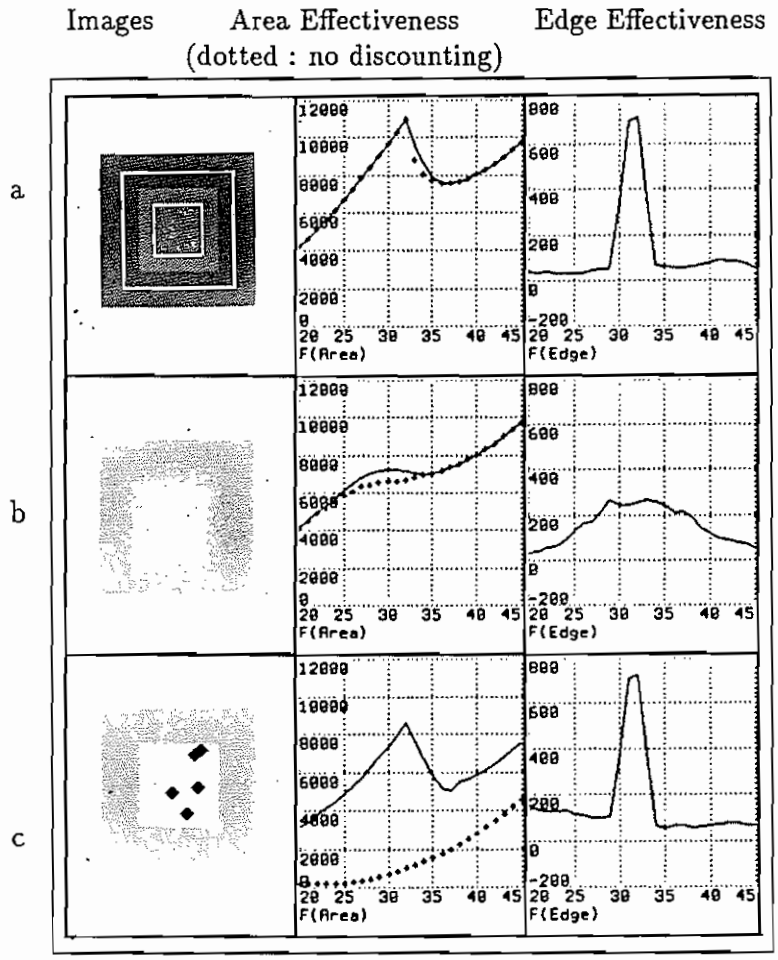


Figure 3: Area and edge effectiveness of a squared patch as a function of their radius. The patches of radius 20 and 45 are outlined on the top image.

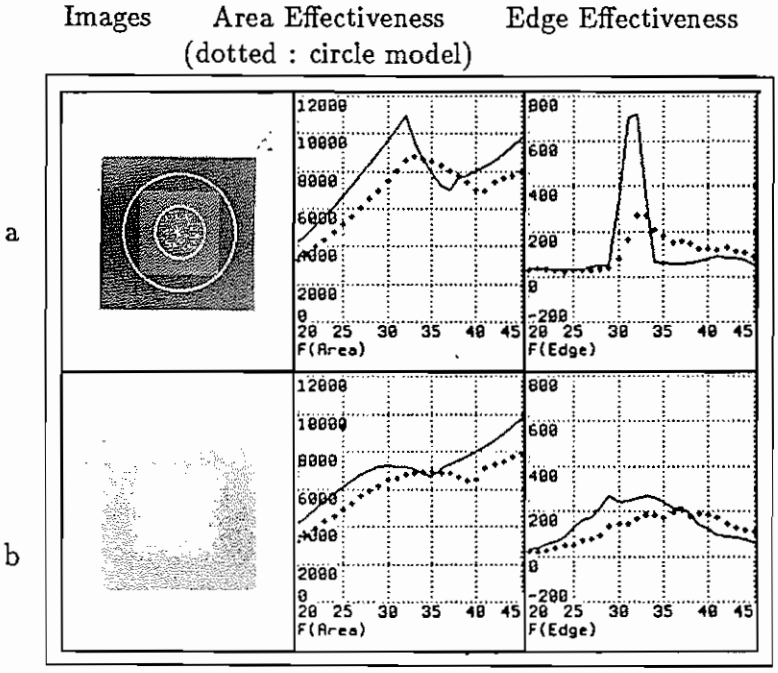
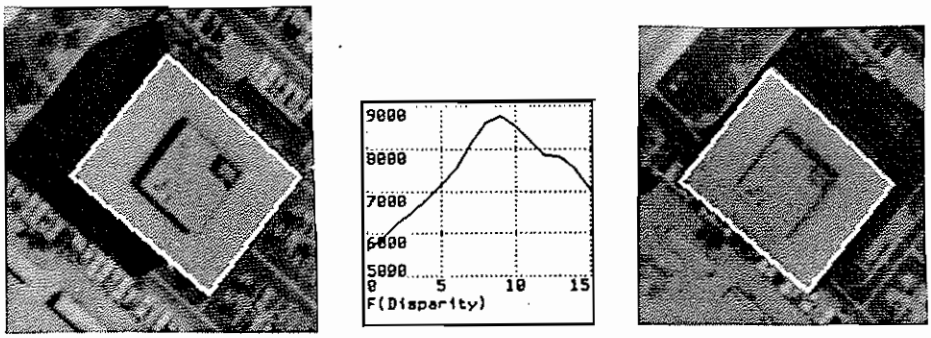


Figure 4: Area and edge effectiveness of a squared patch compared to that of a circular patch as a function of their radius. The circular patches of radius 20 and 45 are outlined on the top image.



(a) (b) (c)

Figure 5: (a) The main rooftop in the left image of Figure 1 (b) F_S as a function of the assumed disparity between left and right image. (c) The projection of the contour in the right image using the best disparity value.

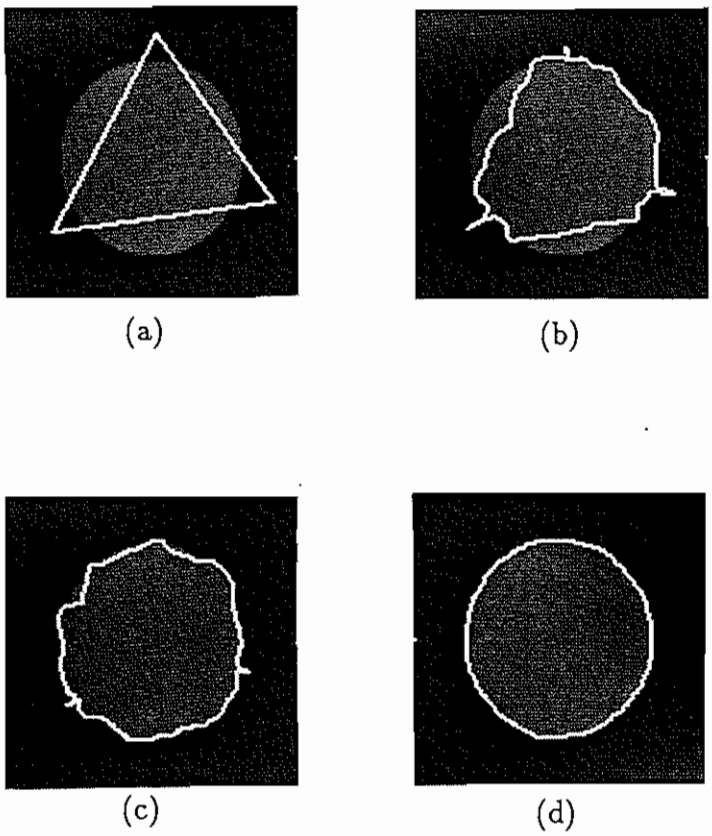


Figure 6: (a) A synthetic image of a circle and the initial position of the curve. (b) (c) The position of the curve after three and seven iterations, respectively. (d) The final outline.

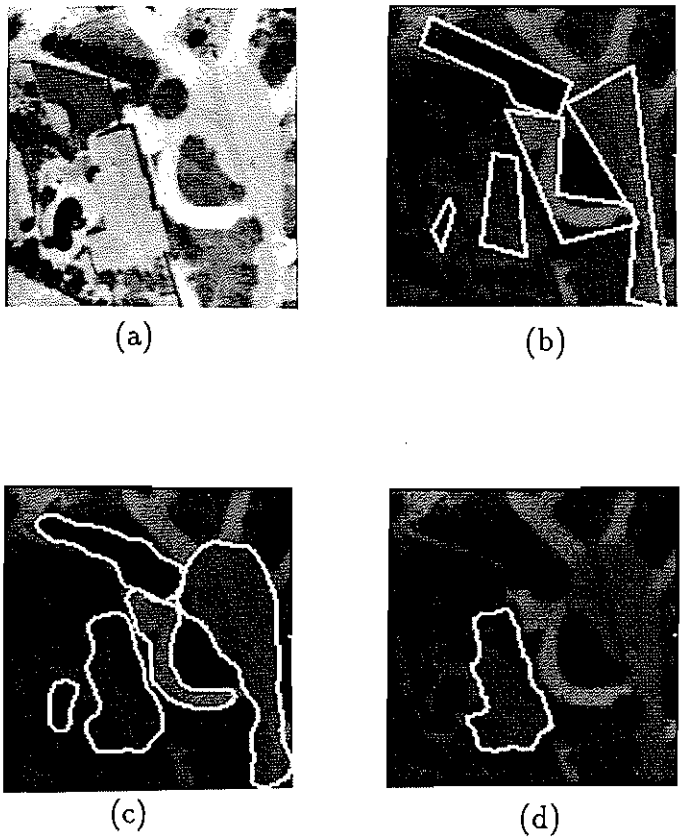
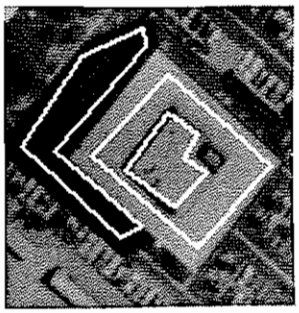
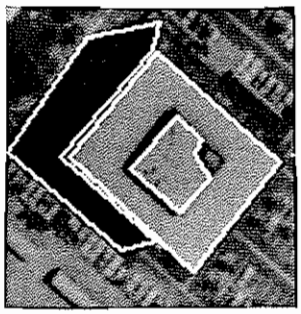


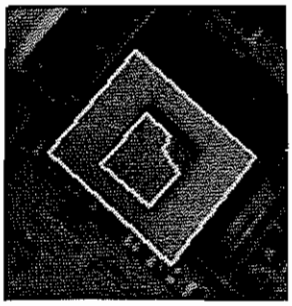
Figure 7: (a) An aerial image of a suburban scene. (b) Interactively entered initial contours. (c) Final outlines after optimization. (d) Outline of the house after reoptimization with less smoothing.



(a)



(b)



(c)

Figure 8: (a) Initial contours in the left image of the pair from Figure 1. (b) Final polygonal outlines after optimization. (c) Matching outlines in the right image.

Segmentation via Minimal-Length Encoding on the Connection Machine

Yvan G. Leclerc

Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

February 17, 1989

Abstract

We present an optimization approach to the image partitioning problem: that of finding a complete and stable description of an image, in terms of a specified descriptive language, that is simplest in the sense of shortest description length. We show that a descriptive language limited to a low-order polynomial description of the intensity variation within each region and a chain-code-like description of the region boundaries yields intuitively satisfying partitions for a wide class of images. By using a regular-grid finite-element representation for the image, the optimization technique, called a continuation method, reduces to a simple, local, parallel, and iterative algorithm that is ideally suited to the Connection Machine.*[tm]

Introduction

Vision can be thought of as a process in which one attempts to infer a description of the outside world, given one or more images of the world, prior information about

*The work reported here was partially supported by the Defense Advanced Research Projects Agency under contract MDA903-86-C-0084.

the world, and prior information about the image sensor. The particular kind of inference process discussed in this paper might be called "inference to the simplest explanation" or, more formally, "minimal-length encoding." The basic idea is that the prior information about the world and the sensor is incorporated in the language used to describe the world and sensor, and the inference process is to find the simplest (i.e.; shortest) description that exactly reproduces the images we are given (see earlier papers [7, 17] for general discussions on this approach). For example, a complete three-dimensional description of the shape and color of objects and light sources, plus a description of the camera parameters would be an extremely efficient description of a large number of images because the only change required from one image to the next would be changes in the camera parameters.

Finding such a three-dimensional description by direct search is impossible, however, because of the exponential number of possible descriptions. Instead, one can imagine a hierarchical descriptive language where, at each level of the hierarchy, one describes the previous level using a language that incorporates more of the three-dimensional nature of the world. With an appropriate choice of incrementally more sophisticated descriptive languages, one can then hope to compute a complete description in a reasonable amount of time.

In this paper, we present an implementation of what might be the first level in such a hierarchy. Specifically, we describe an image as the sum of a piecewise-smooth image and white noise. (A piecewise-smooth image is one that is composed of regions whose attributes are continuous and differentiable up to some specified low order, and for which the region boundaries correspond to discontinuities in these attributes. For this paper, the only attribute we consider is intensity.)

Intuitively, the underlying piecewise-smooth image is meant to model the image we would have obtained if we had used a perfect pin-hole camera, and if the scene had actually been composed of objects with piecewise-smooth surfaces and albedos. The corruption is meant to model both deviations from this idealized piecewise-smooth model of the scene and degradations inherent in image sensors. In particular, we model the corruption as convolution with a known point spread function (to model the point spread function of the lens of a real camera), followed by sampling, quantization, and the addition of white noise (whose variance is unknown and which might also vary in a piecewise-smooth fashion). The white noise is an approximate model of both the deviations from the piecewise-smooth model due to small-scale texturing of the objects (which is why we assume that the variance is not uniform) and sensor noise.

Such a description effectively decomposes the original image into two independent parts. The first part, the underlying image, represents the projection of the many diverse components of the world onto an ideal image plane. Having thus removed the second component, the "noise," we can hope to describe this first part by decomposing it further, using more sophisticated languages.

In the next section, we present in more detail the motivation for posing the inference process as that of finding the simplest description. Next, we present the mathematics and implementation of the special case of a piecewise-constant underlying image. In short, the problem of finding the shortest description is posed as a global optimization problem, wherein the objective function is directly related to the description length. Because of the nonlinearities induced by the discontinuous nature of the underlying image, the objective function is highly nonconvex, so that standard optimization techniques cannot find the global minimum. Instead, a technique called a *continuation method* is used. This technique uses a regular-grid, finite-element representation for the underlying image. With this representation, the continuation method reduces to a simple, local, parallel, and iterative algorithm that is ideally suited to the Connection Machine.^{tm} (Details of the general case are discussed in another paper by this author [10].) Finally, results and timings of the implementation are presented.

Motivation for Simplicity and Stability

Simplicity

The idea that simpler descriptions are better than more complex ones is a strongly intuitive notion that was first enunciated as Occam's razor, which counsels us "not to multiply entities beyond necessity." It reflects not only the intuition that simpler descriptions are better because they are easier to use in many ways, but also the body of scientific and personal experience that tells us there is almost always a simpler description of a set of observations than their mere tabulation.

There are two important assumptions behind this notion. The first assumption is that the data are observations of an underlying structured process, and that we could describe these observations in a much simpler fashion by describing them in terms of that process. The second assumption is that the simpler the description we find, the more likely we are to be describing that underlying process or, at least as far as the observations are concerned, something equivalent to that process.

The idea that simpler is better is quite vague, however: what exactly does it mean for one description to be simpler than another? One possible answer is that the number of degrees of freedom, or of distinct and independent variables in the description, should be the measure of simplicity. Take, for example, the classical curve-fitting problem, in which one is presented with an ordered set of numerical observations that can purportedly be described as points along some mathematically defined curve. The simplest description, then, should be the one that requires the fewest parameters to define the curve. But, even for such a simple problem, one immediately sees that the definition, as stated, is still somewhat vague.

First, the number of parameters required to define a curve depends very much on the vocabulary of curves one brings to bear. For example, if the observations were actually equally spaced points on a quadratic curve, but one attempted to describe them as the sum of sinusoids (as in a discrete Fourier transform), one would require as many parameters as there are observations. However, a polynomial representation would require only six parameters (three specifying the number of observations, spacing and order of the polynomial, and three specifying the coefficients of the polynomial), independently of the number of observations. Thus, one would be inclined to say that the polynomial description is the simpler of the two for these observations.

If, however, one is allowed to use any possible mathematical curve, one must first specify which of the infinite classes of curve the parameters refer to (polynomials versus sinusoids versus ...). That is, we must first specify the *language* in which the description is expressed. Because this clearly requires an infinite number of parameters, one is left with the inescapable conclusion that the vocabulary of curves (or, more generally, the language in which the description is expressed) must be restricted in some sense, or else more parameters than observations will always be needed.

A second fundamental problem posed by this definition of simplicity is that almost all phenomena, and hence observations of them, have an inherent stochastic component. At the very least, the observations will be corrupted in some stochastic manner, even if the underlying phenomenon is purely deterministic. Thus, for our curve-fitting example, even if we could specify the underlying curve with a few variables, we would still need to describe the point-by-point deviations from the curve (either directly or in some appropriate parameter space) to obtain a complete description, and this would require at least as many variables as observations! Again we are left with more variables than observations.

The information-theoretic answer to this quandary is to reduce the idea of an independent variable to its simplest form — a bit. The measure of simplicity then becomes the number of bits in the description that some computationally effective procedure can use to reproduce the observations. This is known as the *minimum-description-length* (MDL) criterion. This criterion, of course, demands prior specification of the computationally effective procedure, which is equivalent to specifying the language in which the description is expressed. Thus, in this formalism, the notion of simplicity is a relative one that depends strongly on the choice of descriptive language.

The necessity of providing an *a priori* descriptive language is a very important and fundamental point. It means that, for a finite number of observations, there is no such thing as an absolute measure of the simplicity of description; simplicity is inescapably a function of one's prior assumptions.

For example, suppose we assume that the underlying process generating the observations in our curve-fitting problem is the sum of a polynomial (of unknown order) and zero-mean white noise (of unknown variance), and that we wish to find the polynomial with the smallest number of nonzero coefficients compatible with this model.

A good descriptive language might then have two components: the first to specify the number of nonzero coefficients and each of their values, and the second to specify the variance and point-by-point values of the added white noise. The curve-fitting problem then becomes that of finding the simplest description (the one with the fewest bits) such that the two components add up exactly to the given observations.

One natural choice for the first component is to assign a fixed number of bits for the specification of the order and for each nonzero coefficient of the polynomial. (The number of bits required is a function of the logarithm of the number of observations, their range, and their precision.) Thus, for this choice of language, polynomials of lower order are simpler to describe than those of higher order.

Because there are provably optimal languages for describing stochastic processes such as white noise,[†] such a language is the natural choice for the second component. With this optimal language, the number of bits required for the second component is roughly proportional to the number of observations times the variance of the point-by-point values.

Thus, with the above descriptive language, there is a natural trade-off between the complexity of the deterministic component (the number of nonzero coefficients) and the complexity of the stochastic component (the variance of the noise): a smaller number of nonzero coefficients reduces the complexity of the first component, but increases the variance of the noise and thus also increases the complexity of the second component; conversely, a larger number of nonzero coefficients increases the complexity of the first component while reducing that of the second.

The image-partitioning problem is similar to the above curve-fitting problem in that each region is described as the sum of an underlying two-dimensional polynomial of unknown order and white noise of unknown variance. Thus, similar languages can be used for this component of the description. In addition, however, we must describe the shape of each region. For this, we use a simple chain code, so that the number of bits is directly proportional to the length of the region boundary. This will be described in more detail in the next section.

The MDL criterion is a significantly more general approach than that of regularization theory [15]. Regularization theory deals with so-called *ill-posed* problems (inverse problems that do not have a unique solution) by adding a measure of the solution's *smoothness*. In the MDL approach, smoothness is only one of many possible measures of simplicity.

[†]An optimal descriptive language is one that minimizes the average number of bits of description per bit of input. This will be discussed in detail shortly.

Stability

The MDL definition of simplicity above is ideal when the descriptive language is optimal for a given class of data. (Formally, a descriptive language is optimal when the sequence of bits in the description is incompressible, so that it is indistinguishable from a completely random sequence of zeroes and ones.) However, describing an image as the corruption of a piecewise-smooth image is clearly suboptimal because we are not taking advantage of the three-dimensional information that gave rise to that underlying piecewise-smooth image. Yet, we cannot go directly to a language that describes the three-dimensional world because of the enormous search space that would be involved.

Because of this suboptimality, it is necessary to introduce an additional heuristic criterion that we call *stability*, by which we mean that certain parts of the description should be unaffected by small changes in the input data. For the image-partitioning problem, this would mean that the number of regions, their shapes, and the order of the polynomials within each region should be unaffected by small changes in the image. The algorithm we present in the next section balances simplicity of description against stability of description by first finding the most stable aspects of the description.

The Piecewise-Constant Case

For this discussion, we shall only consider in detail the special case in which a real image is the sum of an underlying piecewise-constant image and white noise with known variance. The more general case of an underlying piecewise-smooth image and white noise with unknown variance is treated elsewhere [10].

We denote the real $n \times m$ image by the vector $\mathbf{z}$ indexed by $i \in I = 1, \ldots, nm$. The underlying image $u(x, y)$ is represented by a regular grid of square 1×1 elements, with each element centered at the coordinate (x_i, y_i) of the i^{th} pixel in the real image. The 1×1 square centered at (x_i, y_i) is the *spatial domain* $\mathcal{X}_i$ of the i^{th} element, and the value of the element is u_i. Thus,

$$u(x, y) = u_i \qquad \forall\, (x, y) \in \mathcal{X}_i,\ i \in I,$$

and the underlying image is completely represented by the vector $\mathbf{u} = \{u_i,\ i \in I\}$.

Similarly, we represent the noise by the vector $\mathbf{r}$. Thus, the statement that the real image is the sum of the underlying image and the noise can be written as

$$\mathbf{z} = \mathbf{u} + \mathbf{r}. \tag{1}$$

A consequence of this choice of representations is that discontinuities in the underlying image can occur only along the vertical and horizontal boundaries between the grid

elements. One advantage of this is that the underlying image is uniquely specified when there is no noise (namely, $\mathbf{u} = \mathbf{z}$). However, a more sophisticated representation in which elements have variable shape is also possible. This is an excellent avenue for future research.

Using the above definitions, the problem of finding the simplest description is therefore

$$(\mathbf{u}^*, \mathbf{r}^*) = \min_{(\mathbf{u},\mathbf{r}):\, \mathbf{z}=\mathbf{u}+\mathbf{r}} |\mathcal{L}_u(\mathbf{u})| + |\mathcal{L}_r(\mathbf{r})|,$$

where $\mathcal{L}_u$ and $\mathcal{L}_r$ denote the languages used to describe $\mathbf{u}$ and $\mathbf{r}$. From Eq. 1, the equivalent problem is

$$\mathbf{u}^* = \min_{\mathbf{u}} |\mathcal{L}_u(\mathbf{u})| + |\mathcal{L}_r(\mathbf{z} - \mathbf{u})|.$$

There are two steps involved in solving this problem. First, we must define the languages $\mathcal{L}_u$ and $\mathcal{L}_r$. Second, we must specify a computationally feasible procedure for finding $\mathbf{u}^*$ and for determining the stability of the solution.

Defining Descriptive Languages

The first task, then, is to define a language for describing the underlying piecewise-constant image $\mathbf{u}$. By definition, $\mathbf{u}$ is composed of regions of constant intensity. Thus, for each region, we need specify only the shape and position of the region boundaries and the constant intensity within the region. The region boundaries are described by a chain code of unit-length line segments located between adjacent elements; each line segment corresponds to the boundary between adjacent square grid elements. The number of bits required to describe each region is thus proportional to the number of elements in the chain plus a constant to specify the constant intensity and the first element of the chain. The total number of bits required to specify the underlying image is thus proportional to the number of regions plus the total length of the region boundaries.

Because region boundaries occur only when spatially adjacent elements of $\mathbf{u}$ are different, their total length can be determined locally by counting all adjacent elements (u_i, u_j) that have a nonzero difference and dividing by 2 (because region boundaries will be counted twice this way). Thus, the total length of the region boundaries is

$$\frac{1}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(u_i - u_j)),$$

where

$$\begin{aligned} N_i &= \text{the set of neighbors of the } i^{th} \text{ element} \\ \delta(x) &= \text{the Kronecker delta} = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

When the regions are relatively large, a good approximation to the number of bits required to describe u is thus

$$|\mathcal{L}_u(\mathbf{u})| \approx \frac{b}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(u_i - u_j)), \tag{2}$$

where b is the sum of (1) the number of bits required to encode each element in the chain code and (2) the number of bits required to encode the constant intensity and starting element, divided by the average region-boundary length.

As for describing the noise, the fewest bits required to describe data generated by a stochastic process is the negative base-two logarithm of the probability of observing that data [16]. Because we assume the noise to be uncorrelated,

$$\begin{aligned} |\mathcal{L}_r(\mathbf{r})| \equiv -\log_2 P(\mathbf{r}) &= -\log_2 \prod_{i \in I} P(r_i) \\ &= -\sum_{i \in I} \log_2 P(r_i). \end{aligned}$$

Furthermore, we assume the noise to be quantized white noise, where the elements are drawn from a normal distribution and then quantized to the nearest q, the precision of the pixels in the real image. Thus,

$$\begin{aligned} P(r_i) &= \int_{\lfloor r_i \rfloor_q}^{\lceil r_i \rceil_q} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right) dx \\ &\approx \frac{q}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-r_i^2}{2\sigma^2}\right) \quad \text{when } q < \sigma, \end{aligned} \tag{3}$$

and

$$-\log_2 P(\mathbf{r}) \approx nm\,c + a \sum_{i \in I} \left(\frac{r_i}{\sigma}\right)^2. \tag{4}$$

Thus, for u and r satisfying Eq. 1, an approximation to the total number of bits required to describe u and r is

$$|\mathcal{L}_u(\mathbf{u})| + |\mathcal{L}_r(\mathbf{r})| \approx nm\,c + L(\mathbf{u}),$$

where

$$L(\mathbf{u}) = a \sum_{i \in I} \left(\frac{z_i - u_i}{\sigma}\right)^2 + \frac{b}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(u_i - u_j)). \tag{5}$$

Dropping the additive constant, the minimization problem can thus be written as

$$\mathbf{u}^* = \min_{\mathbf{u}} L(\mathbf{u}).$$

Defining a Computationally Feasible Procedure

The simplest, direct way of finding the global minimum of $L(R)$ is to search through all possible sets of regions, calculating the cost for each set, and choosing the set with the smallest cost. Unfortunately, the number of possible sets of regions grows exponentially with the number of elements of $\mathbf{u}$, rendering such a search completely infeasible. Even dynamic programming-like algorithms require at least the evaluation of the cost for every possible simple region, which is an exponential in nm when n and m are greater than 1, again rendering such a search computationally infeasible.

Furthermore, because of the Kronecker delta term, $L(\mathbf{u})$ has many local minima. Thus, standard descent-based optimization techniques are useless. Also, the simulated-annealing style of algorithms exemplified in Geman and Geman [6] are inappropriate, because the time complexity is much too high for this type of function [2]. Intuitively, the reason that stochastic gradient-descent algorithms are inappropriate for this particular objective function is that the function has extremely narrow (in fact, infinitesimally narrow) valleys, so that even stochastic sampling of the surface provides no guidance for the search.

Instead, I have devised an algorithm that yields something very close or equal to the optimal solution for a large class of inputs. It belongs to a class of optimization techniques generally called continuation methods [5, 21]. This algorithm is similar in spirit to the algorithm described in Blake and Zisserman [3] as the "graduated nonconvexity," or GNC algorithm.

As used here, a continuation method embeds the objective function in a family of functions $L(\mathbf{u}, s)$ for which there is a single local minimum at some large s, and for which the number and position of the local minima converge to those of $L(\mathbf{u})$ as s approaches zero. The steps of the continuation method are straightforward. First, find the unique local minimum $\mathbf{u}^0$ of $L(\mathbf{u}, s^0)$ for some sufficiently large s^0. Then, track the local minimum in $\mathbf{u}$ as a decreasing function of s, as follows. For $s^{t+1} = s^t$, let $\mathbf{u}^{t+1}$ be the result of taking a single step of a descent algorithm, as applied to the objective function $L(\mathbf{u}, s^{t+1})$ started at $\mathbf{u} = \mathbf{u}^t$. When the descent algorithm converges, let $s^{t+1} = r\, s^t$ for some $0 < r < 1$, and repeat until s^t is sufficiently small. For an ideal embedding, there will be no bifurcations along this path, and the value of $\mathbf{u}^t$ for a sufficiently large t (and hence a sufficiently small s^t) will be close or equal to the global minimum of $L(\mathbf{u})$.

The specific embedding used here replaces $\delta(u_i - u_j)$ with an exponential,

$$\delta(u_i - u_j) \rightarrow e_{i,j}(\mathbf{u}, s) \equiv \exp\left(-\frac{(u_i - u_j)^2}{(s\sigma)^2}\right),$$

so that

$$L(\mathbf{u}, s) = a \sum_{i \in I} \left(\frac{z_i - u_i}{\sigma}\right)^2 + \frac{b}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - e_{i,j}(\mathbf{u}, s)). \tag{6}$$

This is an appropriate embedding because

$$\lim_{s \to 0} e_{i,j}(\mathbf{u}, s) = \delta(u_i - u_j)$$

so that

$$\lim_{s \to 0} L(\mathbf{u}, s) = L(\mathbf{u}),$$

and, hence, the local minima of $L(\mathbf{u}, s)$ approach the local minima of $L(\mathbf{u})$. Furthermore, there exists a unique local minimum of $L(\mathbf{u}, s)$ for sufficiently large s, namely $\mathbf{u} = \mathbf{z}$. This is so because (1) $L(\mathbf{u}, s) \geq 0 \ \forall \mathbf{u}$, (2) $\mathbf{u} = \mathbf{z}$ is the unique point for which the first summation of Eq. 6 is identically zero, and (3) the second summation vanishes for arbitrarily large s when $\mathbf{u}$ is bounded. Thus, for s approaching infinity, $\mathrm{u} = \mathrm{z}$ is the unique point for which $L(\mathbf{u}, s) = 0$, the unique local (and global) minimum.

Intuitively, the exponential term introduces broad valleys when s is large, and converges to the narrow valleys in the limit as s goes to zero. Thus, the continuation method creates a kind of "scale space" representation of the objective function $L(\mathbf{u})$ (in analogy to Witkin's scale-space representation of a signal [20]) and tracks a local minimum from the coarsest scale (where there is only one local minimum) to the finest scale (where there are many).

Although any iterative descent algorithm can be used for the continuation method (see, for example, the wide variety described in Luenberger's excellent book [12]), the following algorithm has proven to be quite efficient for the objective function examined here. Some experimentation with a conjugate-gradient algorithm has, so far, reduced the number of iterations by only a factor of two, but each step of the algorithm is about twice as long as the simpler one below.

By definition, local minima of $L(\mathbf{u}, s)$ occur when

$$\frac{\partial L(\mathbf{u}, s)}{\partial u_i} = \frac{2a}{\sigma^2}(u_i - z_i) + \frac{2b}{(s\sigma)^2} \sum_{j \in N_i} e_{i,j}(\mathbf{u}, s)(u_i - u_j) = 0, \tag{7}$$

which can be written in vector notation as:

$$\nabla L(\mathbf{u}, s) \equiv \frac{\partial L(\mathbf{u}, s)}{\partial \mathbf{u}} = \mathbf{b} + \mathbf{A}(\mathbf{u}, s)\mathbf{u} = \mathbf{0}, \tag{8}$$

where

$$\begin{aligned}
a_{i,i}(\mathbf{u}, s) &= \frac{2a}{\sigma^2} + \frac{2b}{(s\sigma)^2} \sum_{j \in N_i} e_{i,j}(\mathbf{u}, s) \\
a_{i,j}(\mathbf{u}, s) &= \begin{cases} \dfrac{-2b}{(s\sigma)^2} e_{i,j}(\mathbf{u}, s) & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} \\
b_i &= \frac{-2az_i}{\sigma^2}.
\end{aligned}$$

At each step of the iterative descent algorithm, we linearize the above set of equations by setting $s^{t+1} = rs^t$ and fixing $\mathbf{A}^t \equiv \mathbf{A}(\mathbf{u}^t, s^{t+1})$. Because $\mathbf{A}^t$ is diagonally dominant, a Gauss-Seidel iterate can be used to provide a step in the direction of the solution:

$$u_i^{t+1} = \frac{-1}{a_{i,i}^t}\left(b_i + \sum_{j \neq i} a_{i,j}^t u_j^t\right) = \frac{z_i + \dfrac{b}{a(s^{t+1})^2}\displaystyle\sum_{j \in N_i} e_{i,j}^t u_j^t}{1 + \dfrac{b}{a(s^{t+1})^2}\displaystyle\sum_{j \in N_i} e_{i,j}^t}, \tag{9}$$

where

$$e_{i,j}^t \equiv e_{i,j}(\mathbf{u}^t, s^{t+1}).$$

This is carried out on the Connection Machine^{tm} by assigning each element to a virtual processor in a two-dimensional VP set, and iterating in parallel. The *interaction strengths* $e_{i,j}^t$ are recomputed at each iteration via the NEWS network.

The above is repeated until $|u_i^{t+1} - u_i^t|$ is sufficiently small (less than $0.1 s^{t+1}\sigma$) for all i; only one or two iterations are typically required to achieve this accuracy. Once convergence has been achieved, s is decreased ($s^{t+1} = rs^t$, $0 < r < 1$), and everything repeated until s^{t+1} is sufficiently close to zero.

When the interaction strength falls below $1/e$ (i.e., when $|u_i^t - u_j^t| < s^{t+1}\sigma$), we say that a [tentative] discontinuity between adjacent elements has been found at time t. The discontinuity is called tentative because it is possible (though relatively rare) for the interaction strength to oscillate a few times before converging to a stable value. The word "tentative" will be dropped unless ambiguity would result. The first value of s^{t+1} for which this occurs is called the *stability*, $s_{i,j}$, of the discontinuity.

The reason for calling $s_{i,j}$ a stability measure, as discussed in detail in another paper [10], is that $s_{i,j}$ is approximately equal to the ratio of the local contrast to σ. Thus, when the contrast is sufficiently large relative to σ, the boundary is typically unaffected by small changes to the input image, whereas when the ratio is low, boundaries can shift unpredictably or disappear altogether. Thus, to obtain a stable description, it is necessary to stop the procedure at a reasonably large value of s^{t+1} (typically 1/4 or so). A different strategy might be to stop at a much smaller value, but then use the stability measure in the subsequent stages.

Results

Because of time and graphics software constraints, the figures in this paper were all produced using a Symbolics Lisp-Machine implementation. Similar, but not identical, results were obtained using the Connection Machine.^{tm} The primary reason for the difference is that the Lisp-Machine implementation does not update every element at each Gauss-Seidel iteration. To save time, only those elements that differ significantly

from the previous iteration are updated (except, of course, all elements are updated at the iteration when s^t is decreased). This results in a significant increase in speed on a sequential machine, but also produces a slight degradation in performance. This was not fully appreciated before the Connection Machine^{tm} implementation.

For a 128 $\times$ 128 input image, with a VP ratio of 2 and without floating-point hardware, the Connection Machine^{tm} takes about 0.7 second per iteration for the piecewise-constant case, which is about 100 times faster than the Lisp-Machine implementation (when all elements are updated at each iteration). For the piecewise-first-order case, the time increases to about 3.8 seconds per iteration. In general, the time is approximately $k^2/2$ times the time required for the piecewise-constant case, where k is the number of coefficients in the highest-order polynomial (three for first-order polynomials, six for second order, and so on).

The results presented here were obtained by using the most general form of the encoding-length function, in which the underlying image is piecewise polynomial, the variance of the noise is unknown and piecewise constant, and the sensor model includes a point-spread function. A key point about these examples is that they were all obtained by using *precisely the same parameters*, with the following exceptions. First, a Gaussian point-spread function with $\sigma = 1$ was used for all of the real images, but no point-spread function was used for any of the synthetic images (taking advantage of our *a priori* knowledge about how these synthetic images were created). Second, for demonstrative purposes only and as noted for each example, several values of p_{max}, the order of the underlying image, were used. The conclusion that emerges from these and many other examples not presented here is that a piecewise-second-order underlying image is appropriate for a large class of real images.

The first example illustrates the power of global optimization compared with purely local, noniterative, operations. Figure 1a is the 20 $\times$ 20 input image, which is the sum of a piecewise-first-order image and zero-mean white noise with unit variance. The outer region of the underlying image has intensity 0.0, the center ramp has a slope of 1.0, and the contrast at either end of the ramp with the outer region is 4.0. Of course, the contrast of the center of the ramp with the background is 0.

Figures 1b and 1c illustrate the result of the procedure for $p_{max} = 1$ and 2, respectively, stopping at $s^t = 1/4$. First, note that the entire ramp is separated from the background, even in the center where the local signal-to-noise ratio is 0 (the thinner line separating the ramp from the background near the center indicates that the discontinuity is only of order 1, that is, a discontinuity in the first derivative of the underlying image). This is in contradistinction to the output of the Canny edge detector [4]. For a small spatial scale (Figure 1d), the Canny operator leaves a gap (not to mention the introduction of spurious discontinuities due to the assumption that edges are locally piecewise-constant), whereas a larger spatial scale (Figure 1e) simply makes the artifacts worse. (The operator was unable to find the correct outline for any parameter settings.) Second, note that the elements of the ramp have been determined

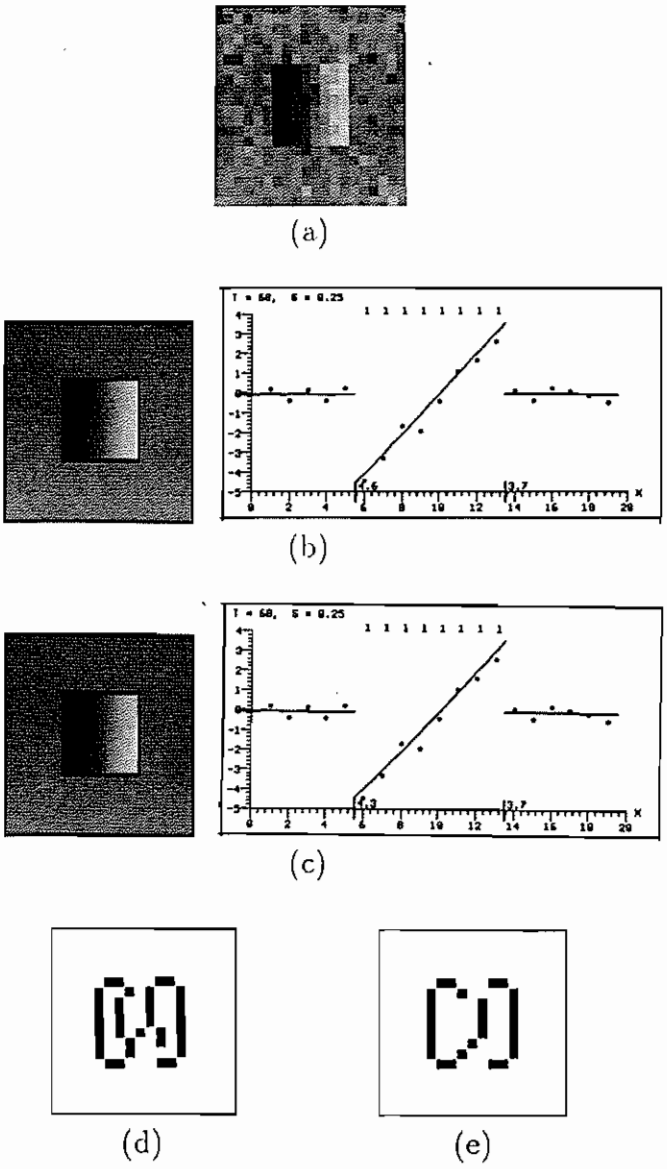


Figure 1: An illustration of the power of global optimization. (a) The input synthetic image. (b) The result of the procedure for $p_{max} = 1$. (c) The result of the procedure for $p_{max} = 2$. (d) The output of the Canny operator, mask size=4. (e) The output of the Canny operator, mask size=8.

to be order 1 (as indicated by the number immediately above each element, no number means that the element is order 0), whereas the elements of the outer region have been determined to be order 0. Thus, the procedure has not only located the discontinuities correctly, but has also determined the correct order for each region.

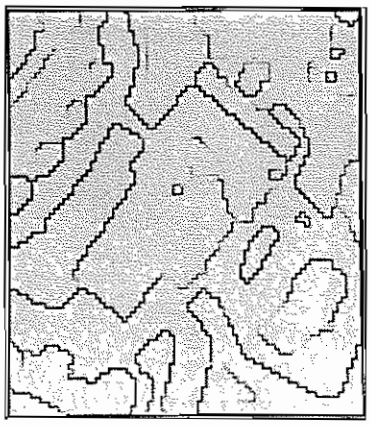
Figure 2 illustrates an application of the procedure to an aerial image of a house, with $p_{max} = 1$, stopping at $s^t = 1/4$. Figures 2b and 2c show the resulting underlying image and discontinuities. Figure 2d is an image of the stability measure for these discontinuities, with the darkest lines indicating the most stable discontinuities. Two interesting points emerge from this example. First, the four bushes in the upper-left corner are almost completely delineated, even though the contrast along that part of their boundaries is virtually nil. This is an example of the "zero contrast" situation similar to the previous synthetic ramp image. Second, the majority of discontinuities that form closed regions have high stability measures. This is a fairly strong indication that the piecewise-first-order (or higher-order) model is appropriate for this image. To verify this conclusion, observe that the discontinuities obtained using $p_{max} = 2$ (Figure 3) are virtually identical, the only exceptions being the few very-low-stability discontinuities.

Figure 4 illustrates an application of the same model with $p_{max} = 1$ (using precisely the same parameters) to the image of a face. In this example, about half the discontinuities have a fairly low stability measure. This indicates that the language is probably not appropriate for this image. This is especially evident in the cheek and chin areas where a higher-order model is clearly more appropriate. Even so, the discontinuities with high stability measures appear to be good candidates for region boundaries. Figure 5 shows the results for $p_{max} = 2$, in which the artifacts due to using too low an order are entirely absent.

Summary

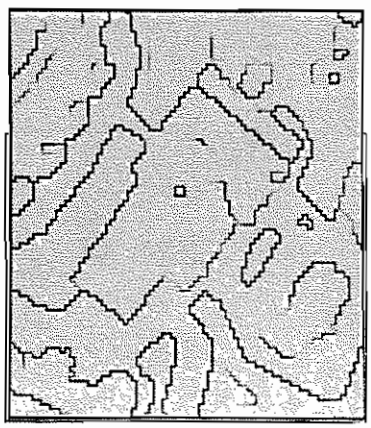
Much work has been done recently on the problem of reconstructing piecewise-smooth surfaces in one or more dimensions, given corrupted samples of the surface [1, 3, 8, 9, 11, 13, 14, 18, 19]. There are several especially difficult aspects to the problem. The first is to determine automatically the appropriate degree of smoothness of the surface as a function of the given data. The second is to determine automatically both the position and order of the discontinuities. The third is to ascertain when such a description is appropriate for the data. We have resolved these difficulties by (1) posing the problem as an optimization problem in which the objective function is based on the information-theoretic notion of minimum-length descriptions, and (2) defining an algorithm that balances simplicity of description against stability of description by first finding the most stable aspects of the description.

(a)



(b) (c) (d)

Figure 2: An application of the procedure to an aerial image of a house, with $p_{max} = 1$. (a) The input image. (b) The resulting underlying image. (c) The underlying image with overlaid discontinuities. (d) The stability measure of the discontinuities; the darkest discontinuities are the most stable.



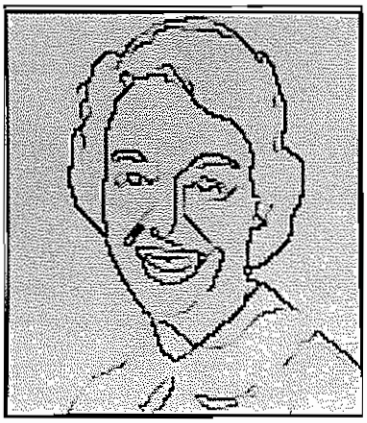
(a) (b) (c)

Figure 3: Same as the prior figure, but with $p_{max} = 2$.

(a)

(b)

(c)

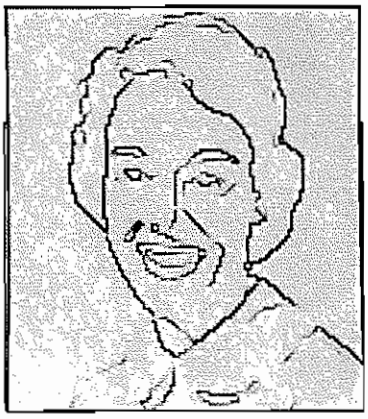


(d)

Figure 4: An application of the procedure to the image of a face, with $p_{max} = 1$. (a) The input image. (b) The resulting underlying image. (c) The underlying image with overlaid discontinuities. (d) The stability measure of the discontinuities.

(a)

(b)



(c)

Figure 5: Same as the prior figure, but with $p_{max} = 2$.

We have presented a new approach to the image-partitioning problem: construct a complete and stable description of an image in terms of a descriptive language that is simplest in the sense of being shortest. We have presented criteria on which to base formal definitions of completeness, stability, and simplicity, and we have embodied these criteria within the theory of minimum-length descriptions. This formalism is very general and is likely to be applicable to other stages of the scene-analysis process.

For the specific image-partitioning problem, we described real images as the corruption of ideal (piecewise-polynomial) images by blurring and the addition of spatially varying white noise. We defined a language for describing both the ideal image and the corruptions, and presented an algorithm for finding the simplest description of an image, in terms of this language, for a given measure of stability. This measure has proved *crucial* because we are interested in descriptions that are not only as simple as possible, but that are also as invariant as possible to the severe approximations embodied in any low-level descriptive language. The algorithm not only determines the position of discontinuities in the ideal image, but also determines both the order of the discontinuity and the order of the polynomial within the regions; all of this is done without the need to adjust any parameters. Furthermore, the algorithm is local, parallel, and iterative, making it ideally suited to massively parallel computer architectures such as the Connection Machine.[tm]

Applications of this formalism to real images indicate that, even though the descriptive language we have defined is extremely simple (with no models of three-dimensional shape, lighting, or texture, for example), the simplest and most stable description in this language yields excellent image partitions.

References

[1] Besl, P.J., and R.C. Jain, "Segmentation Through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10(2), pp. 167–192, 1988.

[2] Blake, A., "Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11(1), pp. 2–12, 1989.

[3] Blake, A., and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.

[4] Canny, J.F., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(6), pp. 679–698, 1986.

[5] Dahlquist, G., and Å. Björck, *Numerical Methods*, N. Anderson (trans.), Prentice Hall, Englewood Cliffs, NJ, 1974.

[6] Geman, S., and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6(6), pp. 721–741, 1984.

[7] Georgeff, M.P., and C.S. Wallace, "A General Selection Criterion for Inductive Inference," *SRI Technical Note* 372, SRI International, Menlo Park, CA, 1985.

[8] Grimson, W.E.L., and T. Pavlidis, "Discontinuity Detection for Visual Surface Reconstruction," *Computer Vision, Graphics, and Image Processing*, Vol. 30, pp. 316–330, 1985.

[9] Langridge, D.J., "Detection of Discontinuities in the First Derivatives of Surfaces," *Computer Vision, Graphics, and Image Processing*, Vol. 27, pp. 291–308, 1984.

[10] Leclerc, Y.G., "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision,* Vol. 2(4), 1988.

[11] Lee, D., and T. Pavlidis, "One-Dimensional Regularization with Discontinuities," in *Proceedings of the First International Conference on Computer Vision,* London, England, pp. 572–577, June 8–11, 1987.

[12] Luenberger, D.G., *Linear and Nonlinear Programming* (second edition), Addison-Wesley, Menlo Park, CA, 1984.

[13] Marroquin, J., S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *Journal of the American Statistical Association*, Vol. 82(397), pp. 76–89, 1987.

[14] Mumford, D., and J. Shah, "Boundary Detection by Minimizing Functionals, I," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* San Francisco, CA, pp. 22–26, 1985.

[15] Poggio, T., V. Torre, and C. Koch, "Computational Vision and Regularization Theory," *Nature*, Vol. 317, 1985.

[16] Rissanen, J., "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics*, Vol. 11(2), pp. 416–431, 1983.

[17] Rissanen, J., "Minimum-Description-Length Principle," *Encyclopedia of Statistical Sciences*, J. Wiley, NY, Vol. 5, pp. 523–527, 1987.

[18] Saint-Marc, P., and G. Medioni, "Adaptive Smoothing for Feature Extraction," in *Proceedings of the DARPA Image Understanding Workshop*, Boston, MA, pp. 1100–1113, April, 1988.

[19] Terzopoulos, D., "Regularization of Inverse Visual Problems Involving Discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(4), pp. 413–424, 1986.

[20] Witkin, A.W., "Scale Space Filtering," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, pp. 1019–1021, 1983.

[21] Witkin, A.W., D. Terzopoulos, and M. Kass, "Signal Matching Through Scale Space," *International Journal of Computer Vision*, Vol. 1(2), pp. 133–144, 1987.